

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Hai Jin Yi Pan Nong Xiao
Jianhua Sun (Eds.)

Grid and Cooperative Computing – GCC 2004 Workshops

GCC 2004 International Workshops
IGKG, SGT, GISS, AAC-GEVO, and VVS
Wuhan, China, October 21-24, 2004
Proceedings



Springer

Volume Editors

Hai Jin

Jianhua Sun

Huazhong University of Science and Technology, Cluster and Grid Computing Lab
430074 Wuhan, China

E-mail: {hjin, jhsun}@hust.edu.cn

Yi Pan

Georgia State University, Department of Computer Science

34 Peachtree Street, Suite 1450, Atlanta, GA 30302-4110, USA

E-mail: pan@cs.gsu.edu

Nong Xiao

National University of Defense Technology, School of Computer

Changsha, 410073 China

E-mail: xiao-n@vip.sina.com

Library of Congress Control Number: 2004113699

CR Subject Classification (1998): C.2, D.4, I.2.11, H.4, H.3, H.5, K.6.5

ISSN 0302-9743

ISBN 3-540-23578-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik

Printed on acid-free paper SPIN: 11323686 06/3142 5 4 3 2 1 0

Preface

Welcome to the proceedings of GCC 2004 and the city of Wuhan. Grid computing has become a mainstream research area in computer science and the GCC conference has become one of the premier forums for presentation of new and exciting research in all aspects of grid and cooperative computing. The program committee is pleased to present the proceedings of the 3rd International Conference on Grid and Cooperative Computing (GCC 2004), which comprises a collection of excellent technical papers, posters, workshops, and keynote speeches. The papers accepted cover a wide range of exciting topics, including resource grid and service grid, information grid and knowledge grid, grid monitoring, management and organization tools, grid portal, grid service, Web services and their QoS, service orchestration, grid middleware and toolkits, software glue technologies, grid security, innovative grid applications, advanced resource reservation and scheduling, performance evaluation and modeling, computer-supported cooperative work, P2P computing, automatic computing, and meta-information management.

The conference continues to grow and this year a record total of 581 manuscripts (including workshop submissions) were submitted for consideration. Expecting this growth, the size of the program committee was increased from 50 members for GCC 2003 for 70 in GCC 2004. Relevant differences from previous editions of the conference: it is worth mentioning a significant increase in the number of papers submitted by authors from outside China; and the acceptance rate was much lower than for previous GCC conferences. From the 427 papers submitted to the main conference, the program committee selected only 96 regular papers for oral presentation and 62 short papers for poster presentation in the program. Five workshops, International Workshop on Agents, and Autonomic Computing, and Grid Enabled Virtual Organizations, International Workshop on Storage Grids and Technologies, International Workshop on Information Security and Survivability for Grid, International Workshop on Visualization and Visual Steering, International Workshop on Information Grid and Knowledge Grid, complemented the outstanding paper sessions.

The submission and review process worked as follows. Each submission was assigned to three program committee members for review. Each program committee member prepared a single review for each assigned paper or assigned a paper to an outside reviewer for review. Given the large number of submissions, each program committee member was assigned roughly 15–20 papers. The program committee members consulted 65 members of the grid computing community in preparing the reviews. Based on the review scores, the program chairs made the final decision. Given the large number of submissions, the selection of papers required a great deal of work on the part of the committee members.

Putting together a conference requires the time and effort of many people. First, we would like to thank all the authors for their hard work in preparing submissions to the conference. We deeply appreciate the effort and contributions of the program committee members who worked very hard to select the very best submissions and to put together an exciting program. We are also very grateful for the numerous suggestions

we received from them. Also, we especially thank the effort of those program committee members who delivered their reviews in a timely manner despite having to face very difficult personal situations. The effort of the external reviewers is also deeply appreciated. We are also very grateful to Ian Foster, Jack Dongarra, Charlie Catlett, and Tony Hey for accepting our invitation to present a keynote speech, and to Depei Qian for organizing an excellent panel on a very exciting and important topic. Thanks go to the workshop chairs for organizing five excellent workshops on several important topics in grid computing. We would also like to thank Pingpeng Yuan for installing and maintaining the submission website and working tirelessly to overcome the limitations of the tool we used.

We deeply appreciate the tremendous efforts of all the members of the organizing committee. We would like to thank the general co-chairs, Prof. Andrew A. Chien and Prof. Xicheng Lu for their advice and continued support. Finally, we would like to thank the GCC steering committee for the opportunity to serve as the program chairs as well as their guidance through the process. We hope that the attendees enjoyed this conference and found the technical program to be exciting.

Hai Jin and Yi Pan

Conference Committees

Steering Committee

Guojie Li (Institute of Computing Technology, CAS, China)
Xiaodong Zhang (National Science Foundation, USA)
Zhiwei Xu (Institute of Computing Technology, CAS, China)
Xianhe Sun (Illinois Institute of Technology, USA)
Jun Ni (University of Iowa, USA)
Hai Jin (Huazhong University of Science and Technology, China)
Minglu Li (Shanghai Jiao Tong University, China)

Conference Co-chairs

Andrew A. Chien (University of California at San Diego, USA)
Xicheng Lu (National University of Defense Technology, China)

Program Co-chairs

Yi Pan (Georgia State University, USA)
Hai Jin (Huazhong University of Science and Technology, China)

Workshop Chair

Nong Xiao (National University of Defense Technology, China)

Panel Chair

Depei Qian (Xi'an Jiaotong University, China)

Publicity Chair

Minglu Li (Shanghai Jiao Tong University, China)

Tutorial Chair

Dan Meng (Institute of Computing Technology, CAS, China)

Poster Chair

Song Wu (Huazhong University of Science and Technology, China)

Program Committee Members

Mark Baker (University of Portsmouth, UK)
Rajkumar Buyya (University of Melbourne, Australia)
Wentong Cai (Nanyang Technological University, Singapore)
Jiannong Cao (Hong Kong Polytechnic University, Hong Kong)
Guihai Chen (Nanjing University, China)
Xiaowu Chen (Beihang University, China)
Xuebin Chi (Computer Network Information Center, CAS, China)
Qianni Deng (Shanghai Jiao Tong University, China)
Shoubin Dong (South China University of Technology, China)
Xiaoshe Dong (Xi'an Jiaotong University, China)
Dan Feng (Huazhong University of Science and Technology, China)
Ning Gu (Fudan University, China)
Yadong Gui (Shanghai Supercomputer Center, China)
Minyi Guo (University of Aizu, Japan)
Yanbo Han (Institute of Computing Technology, CAS, China)
Yanxiang He (Wuhan University, China)
Jinpeng Huai (Beihang University, China)
Chun-Hsi Huang (University of Connecticut, USA)
Liusheng Huang (University of Science and Technology of China, China)
Kai Hwang (University of Southern California, USA)
Weijia Jia (City University of Hong Kong, Hong Kong)
Francis Lau (The University of Hong Kong, Hong Kong)
Keqin Li (State University of New York, USA)
Minglu Li (Shanghai Jiao Tong University, China)
Qing Li (City University of Hong Kong, Hong Kong)
Qinghua Li (Huazhong University of Science and Technology, China)
Xiaoming Li (Peking University, China)
Xiaola Lin (City University of Hong Kong, Hong Kong)
Xinda Lu (Shanghai Jiao Tong University, China)
Zhengding Lu (Huazhong University of Science and Technology, China)
Junzhou Luo (Southeast University, China)
Dan Meng (Institute of Computing Technology, CAS, China)
Xiangxu Meng (Shandong University, China)
Xiaofeng Meng (Renmin University of China, China)
Geyong Min (University of Bradford, UK)
Jun Ni (University of Iowa, USA)
Lionel Ni (Hong Kong University of Science and Technology, Hong Kong)
Depei Qian (Xi'an Jiaotong University, China)
Yuzhong Qu (Southeast University, China)

Hong Shen (Japan Advanced Institute of Science and Technology, Japan)
Ke Shi (Huazhong University of Science and Technology, China)
Ninghui Sun (Institute of Computing Technology, CAS, China)
Yuzhong Sun (Institute of Computing Technology, CAS, China)
David Taniar (Monash University, Australia)
Huanglory Tianfield (Glasgow Caledonian University, UK)
Weiqin Tong (Shanghai University, China)
David W. Walker (Cardiff University, UK)
Cho-Li Wang (The University of Hong Kong, Hong Kong)
Xingwei Wang (Northeastern University, China)
Jie Wu (Florida Atlantic University, USA)
Song Wu (Huazhong University of Science and Technology, China)
Zhaohui Wu (Zhejiang University, China)
Nong Xiao (National University of Defense Technology, China)
Cheng-Zhong Xu (Wayne State University, USA)
Baoping Yan (Computer Network Information Center, CAS, China)
Guangwen Yang (Tsinghua University, China)
Laurence Tianruo Yang (St. Francis Xavier University, Canada)
Qiang Yang (Hong Kong University of Science and Technology, Hong Kong)
Shoubao Yang (University of Science and Technology of China, China)
Zhonghua Yang (Nanyang Technological University, Singapore)
Pingpeng Yuan (Huazhong University of Science and Technology, China)
Weimin Zheng (Tsinghua University, China)
Yao Zheng (Zhejiang University, China)
Luo Zhong (Wuhan University of Technology, China)
Aoying Zhou (Fudan University, China)
Wanlei Zhou (Deakin University, Australia)
Xinrong Zhou (Åbo Akademi University, Finland)
Jianping Zhu (University of Akron, USA)
Mingfa Zhu (Lenovo Research, China)
Hai Zhuge (Institute of Computing Technology, CAS, China)

Local Arrangements Chair

Pingpeng Yuan (Huazhong University of Science and Technology, China)

Exhibition Chair

Qin Zhang (Huazhong University of Science and Technology, China)

Financial Chair

Xin Li (Huazhong University of Science and Technology, China)

Industry Chair

Xia Xie (Huazhong University of Science and Technology, China)

Publication Chair

Jianhua Sun (Huazhong University of Science and Technology, China)

Conference Secretary

Cong Geng (Huazhong University of Science and Technology, China)

Reviewers

Rashid Al-Ali	Yingwei Luo	Xianbing Wang
Jeff Dallien	Wendy MacCaul	Xiaofang Wang
Zhiqun Deng	Praveen Madiraju	Xiaolin Wang
Jonathan Giddy	Shalil Majithia	Xingwei Wang
Ian Grimstead	Zhongquan Mao	Yuelong Wang
Zhengxiong Hou	Stephen Pellicer	Mark Wright
Yanli Hu	Weizong Qiang	Guang Xiang
Ajay Katangur	Ling Qiu	Bin Xiao
Yunchun Li	Shrija Rajbhandari	Xia Xie
Na Lin	Omer Rana	Shaomin Zhang
Zhen Lin	Geoffrey Shea	Yang Zhang
Hui Liu	Praveena Tayanthi	Ran Zheng
Tao Liu	Ian Taylor	Jingyang Zhou
Xinpeng Liu	Baoyi Wang	Cheng Zhu
Sanglu Lu	Guojun Wang	Deqing Zou
Zhongzhi Luan	Hui Wang	

Table of Contents

Workshop 1: International Workshop on Information Grid and Knowledge Grid (IGKG'2004)

ART Based Predictive Caching System for XML P2P Database	3
<i>Wan-Song Zhang and Da-Xin Liu</i>	
DART-FAS: Federated Access Service on Database Grid	11
<i>Guozhou Zheng, Zhaohui Wu, and Chang Huang</i>	
A Scalable Information Grid Architecture Based on P2P and Web Service Technologies	19
<i>Dugki Min and Eunmi Choi</i>	
Research on Service-Oriented Software Framework	27
<i>Ying Li, Zhaohui Wu, and Shuiguang Deng</i>	
An Accounting and QoS Model for Grid Computing	36
<i>Yang Guangwen, Yongwei Wu, Dazheng Huang, and Weimin Zheng</i>	
Reputation-Aware Contract-Supervised Grid Computing	44
<i>Xiangli Qu, Nong Xiao, Guang Xiang, and Xue-Jun Yang</i>	
The Analysis of Efficient Monitoring Grid Traffic with Flow Conservation Equation	52
<i>Xianghui Liu, Jianping Yin, Xicheng Lu, Zhiping Cai, and Jianmin Zhao</i>	
Dynamic Semantic Clustering Approach for Web User Interest	59
<i>Jiu Jun Chen, Ji Gao, Bei Shui Liao, and Jun Hu</i>	
Building Interoperable Software Components Repository Based on MMF	67
<i>Bing Li, Keqing He, Jin Liu, Qiang Wang, Peng Liang, and Rong Peng</i>	
An Approach for Constructing Software Component Repository in Grid Environments	75
<i>Dehui Du, Shi Ying, Keqing He, and Yangfan He</i>	
Agent-Based Modeling for Virtual Organizations in Grid	83
<i>Yuqing Zhai, Yuzhong Qu, and Zhiqiang Gao</i>	
A Pattern-Based Approach to Facilitating Service Composition	90
<i>Haitao Hu, Yanbo Han, Kui Huang, Gang Li, and Zhuofeng Zhao</i>	

Automatic Service Matching and Service Discovery Based on Ontology	99
<i>Chuan Lin, Zhaohui Wu, Shuiguang Deng, and Li Kuang</i>	
An Algorithm for Calculating Process Similarity to Cluster Open-Source Process Designs	107
<i>Kui Huang, Zhaotao Zhou, Yanbo Han, Gang Li, and Jing Wang</i>	
A Study on Semantic Web-Based Credit Evaluation Service	115
<i>Jing Fan, Bo Ren, and Jia-Mei Cai</i>	
Research on System Architecture and Service Composition of Spatial Information Grid	123
<i>Yu Tang and Ning Jing</i>	
Open Language Approach for Dynamic Service Evolution	132
<i>Thomas Weishäupl and Erich Schikuta</i>	
Adaptive Grid Workflow Scheduling Algorithm	140
<i>Shaohua Zhang, Yujin Wu, and Ning Gu</i>	
Building a Courseware Grid upon Dart Database Grid	148
<i>Meiyu Fang, Zhaohui Wu, Huajun Chen, Yuxin Mao, Guozhou Zheng, Zhao Xu, and Xiaojun Wu</i>	
A Novel Agent-Based Load Balancing Algorithm for Grid Computing	156
<i>Shudong Chen, Wenju Zhang, Fanyuan Ma, Jianhua Shen, and Minglu Li</i>	
Virtual Battlefield Attack-Defense Countermeasure Simulation on the Grid	164
<i>Jianhua Yang, Zhaohui Wu, Siliang Tang, and Xiaosheng Guo</i>	
Virtual Semantic Resource Routing Algorithm for Multimedia Information Grid	173
<i>Haiyang Zhang and Huadong Ma</i>	
Digital Library Application Grid – An Opportunity to Open Cultural Infrastructure	181
<i>Zhongzhi Luan, Depei Qian, Xiaoshe Dong, Xingjun Zhang, and Yunchun Li</i>	
Workshop 2: International Workshop on Storage Grid and Technologies (SGT'2004)	
A JDO Storage Cluster Based on Object Devices	187
<i>Youhui Zhang, Dongsheng Wang, Chongnan Gao, and Weimin Zheng</i>	
iNASC: A iSCSI-Based NAS Storage Cluster	195
<i>Dezhi Han, Changsheng Xie, and Faling Yi</i>	

Design and Implementation of a Non-volatile RAM Disk in the SAN Environment	203
<i>Ji-wu Shu, Bing Yu, and Rui Yan</i>	
Engineering Web Storage Servers Using Session Management	213
<i>Min Qu, Yafei Dai, and Yang Zhao</i>	
Topology and Resource Discovery in Peer-to-Peer Overlay Networks	221
<i>Dongsheng Li, Nong Xiao, and Xicheng Lu</i>	
An Implementation of Semi-synchronous Remote Mirroring System for SANs	229
<i>Rui Yan, Ji-wu Shu, and Dong-chan Wen</i>	
A Security Scheme for United Storage Network	238
<i>Yihui Luo, Changsheng Xie, and Chengfeng Zhang</i>	
STS: A Share Taper System for Storage Area Networks	246
<i>Jiefeng Xu and Zheng Qin</i>	
Storage Virtualization System with Load Balancing for SAN	254
<i>Weitao Sun, Ji-wu Shu, and Weimin Zheng</i>	
Design and Optimization of an iSCSI System	262
<i>Bigang Li, Ji-wu Shu, and Weimin Zheng</i>	
Measurement and Modeling of Large-Scale Peer-to-Peer Storage System	270
<i>Gang Liu, Mingzeng Hu, Binxing Fang, and Hongli Zhang</i>	
A Virtual Tape System Based on Storage Area Networks	278
<i>Fei Mu, Ji-wu Shu, Bigang Li, and Weimin Zheng</i>	
Cocktail Search in Unstructured P2P Networks	286
<i>Xiuguo Bao, Binxing Fang, and Mingzeng Hu</i>	
Data I/O Optimization in Storage Systems	294
<i>Di Wang, Ji-wu Shu, and Meiming Shen</i>	
Data Resource Discovery in a Computational Grid	303
<i>Sajindra Jayasena, Chin-Peng Yee, Jie Song, Abele Stoelwinder, Chong Wee See, and Wai-Hong Wong</i>	
The Design and Implementation of a Locking Mechanism for a Distributed Com- puting Environment	311
<i>Jaechun No, Hyo Kim, and Jang-sun Lee</i>	
Replica Location Mechanism Based on DHT and the Small-World Theory	319
<i>Xindong You, Guiran Chang, Wei Yang, Wandan Zeng, and Xueyao Chen</i>	

Workshop 3: International Workshop on Information Security and Survivability for Grid (GISS'2004)

A New Chameleon Multi-signature Based on Bilinear Pairing	329
<i>Chunbo Ma and Dake He</i>	
A Topology-Adapted Network Defense Model Based on Mobile Agent	335
<i>Yichuan Jiang, Yiping Zhong, and Shiyong Zhang</i>	
AT-RBAC: An Authentication Trustworthiness-Based RBAC Model	343
<i>Lunwei Wang, Lifeng Wei, Xiangke Liao, and Huaimin Wang</i>	
WBEM Based Distributed Network Monitoring	351
<i>Bo Liu and Hui Liu</i>	
Multiparty Joint Authentication: Extending the Semantics of Single Sign-On for Grids	358
<i>Hui Liu and Minglu Li</i>	
A Software Engineering Perspective for Services Security	366
<i>Jun Han</i>	
Algorithms for Congestion Detection and Control	374
<i>Wu Liu, Hai-Xin Duan, Jian-Ping Wu, Xing Li, and Ping Ren</i>	
Modeling Time-Related Trust	382
<i>Chenlin Huang, Hua-Ping Hu, and Zhiying Wang</i>	
Defending DDoS Attacks Using Network Traffic Analysis and Probabilistic Packet Drop	390
<i>Jungtaek Seo, Cheolho Lee, and Jongsub Moon</i>	
Building a Secure Infrastructure for P2P Applications in Mobile Ad Hoc Networks	398
<i>Guangming Hu, Zunguo Huang, Hua-Ping Hu, and Zhenghu Gong</i>	
Middleware Framework for Secure Grid Application in Mobile Web Services Environment	406
<i>Namje Park, Kiyoun Moon, Jongsu Jang, and Sungwon Sohn</i>	
Autonomic Computing for Defense-in-Depth Information Assurance: Architecture and a Case Study	414
<i>Xin Xu, Zunguo Huang, and Lei Xuan</i>	
Mining Maximal Frequent Itemsets for Intrusion Detection	422
<i>Hui Wang, Qing-Hua Li, Huanyu Xiong, and Sheng-Yi Jiang</i>	

Context-Aware Role-Based Access Control Model for Web Services	430
<i>Xu Feng, Xie Jun, Huang Hao, and Xie Li</i>	
Policy-Tree Based Proactive Defense Model for Network Security	437
<i>Feng Zhang, Zhiguang Qin, and Shijie Zhou</i>	
Researches on Scalable Architecture for Security Information Distribution Service	450
<i>Haitao Chen, Chuanfu Xu, Zunguo Huang, Zhenghu Gong, and Hua-Ping Hu</i>	
A Sequential Pattern Mining Algorithm for Misuse Intrusion Detection	458
<i>Shi-Jie Song, Zunguo Huang, Hua-Ping Hu, and Shi-Yao Jin</i>	
Network Performance Measurement Methodologies in PGMS	466
<i>Yuanzhe Yao, Binxing Fang, Hongli Zhang, and Wei Wang</i>	
Via Firewalls	474
<i>Lu Yan</i>	
A Grid Security Infrastructure Based on Behaviors and Trusts	482
<i>Xiaolin Gui, Bing Xie, Yinan Li, and Depei Qian</i>	
Research on a Quantitative Security Risk Assessment Approach in Large-Scale Early Warning System	490
<i>Lei Xuan and Xin Xu</i>	
A Formal Logic for Shared Resource Access Control in the Grid	498
<i>Baiyan Li, Ruonan Rao, Jinyuan You, and Minglu Li</i>	
Security Enhanced to GSI: An Integrated Framework with a Mechanism	506
<i>Baoliang Zhang, Hanping Hu, Xiaogang Wu, and Tao Kong</i>	
Reliable Accounting in Grid Economic Transactions	514
<i>Luigi Catuogno, Pompeo Faruolo, Umberto Ferraro Petrillo, and Ivan Visconti</i>	
A Gravity-Based Intrusion Detection Method	522
<i>Sheng-Yi Jiang, Qing-Hua Li, and Hui Wang</i>	
Anomaly Detection Using Fast SOFM	530
<i>Jun Zheng, Mingzeng Hu, Binxing Fang, and Hongli Zhang</i>	
Research on Secure Multicast Technology in Grid-Based Large-Scale Distributed Simulation Applications	538
<i>Wei Wu, Ling Shen, Haitao Huo, and Xiaojian Li</i>	

BSCM: Proposed Security Collaboration Model Based on Blackboard	546
<i>Yong Tang, Hua-Ping Hu, and Xicheng Lu</i>	
Credential Trustworthiness-Based Trust Evaluation in Grid	554
<i>Xiangli Qu, Xue-Jun Yang, Guang Xiang, and Haifang Zhou</i>	
Workshop 4: International Workshop on Agents and Autonomic Computing and Grid Enabled Virtual Organizations (AAC-GEVO'2004)	
Towards the Merger of Grid and Economy	563
<i>Thomas Weishäupl and Erich Schikuta</i>	
An Approach to Evaluate Communication Cost in Grid System	571
<i>DaDong Wang, HongJun Wang, RuiJun Wang, HaiYing Liang, and Yuan Gao</i>	
Towards the Automation of Autonomic Systems	579
<i>Walid Chainbi</i>	
A Coverage-Preserving Node Scheduling Algorithm for Self-organized Wireless Sensor Networks	587
<i>Jie Jiang and Wenhua Dou</i>	
Evaluation Issues in Autonomic Computing	597
<i>Julie A. McCann and Markus C. Huebscher</i>	
Multi-path QoS Routing in TDMA/CDMA Ad Hoc Wireless Networks	609
<i>Huayi Wu, Chuanhe Huang, Xiaohua Jia, and Yanxiang He</i>	
Economic Heuristic Guided Price-Regulating Mechanism in SHGRB	617
<i>Tang Jun, Tong Weiqin, Zhi Xiaoli, and Yin Zhijie</i>	
Workflow-Based Grid Portal for Quantum Mechanics	625
<i>Sung-Wook Byun, Yun-Kyoung Lee, Yong-Won Kwon, So-Hyun Ryu, and Chang-Sung Jeong</i>	
Agent-Oriented Formal Specification of Web Services	633
<i>Hong Zhu, Bin Zhou, Xinjun Mao, Lijun Shan, and David Duce</i>	
Autonomic Incident Manager for Enterprise Applications	642
<i>Renuka Sindhgatta, Swaminathan Natarajan, Krishnakumar Pooloth, Colin Pinto, and N.S. Nagaraj</i>	
HiGAF: A Hierarchical Grid Accounting Framework	650
<i>Qi Qian and Minglu Li</i>	
Agent-Based Resource Selection for Grid Computing	658
<i>Song Shen and Gregory M.P. O'Hare</i>	

Workshop 5: International Workshop on Visualization and Visual Steering (VVS'2004)

Visualization Practice in Construction of a Computational Grid for Multidisciplinary Design Optimization of Aero-crafts	673
<i>Hong Liu, Xin-hua Lin, Xi-li Sun, Xiao-Feng Qi, Qian-ni Deng, and Xin-da Lu</i>	
EEMAS: An Enabling Environment for Multidisciplinary Application Simulations	681
<i>Lijun Xie, Yao Zheng, Jifa Zhang, Xin Huang, and Zhengge Huang</i>	
Services for Parallel Remote-Sensing Image Processing Based on Computational Grid	689
<i>Xue-Jun Yang, Zhi-ming Chang, Haifang Zhou, Xiangli Qu, and Chun-Jiang Li</i>	
Configuration of the Galaxy Grid Node Environment	697
<i>Guoteng Pan, Qiang Dou, Luoguo Xie, Guangming Liu, Haifang Zhou, Xi- aoqian Zhu, and Erhua He</i>	
GVis: A Java-Based Architecture for Grid Enabled Interactive Visualization	704
<i>Youbing Zhao, Wei Chen, Yingjun Qiu, and Jiaoying Shi</i>	
Balancing CPU and GPU: Real-Time Visualization of Large Scale 3D Scanning Models	712
<i>Zhao Dong, Wei Chen, Long Zhang, and Qunsheng Peng</i>	
LRZB, a Hybrid Algorithm of Local Ray-Casting and Z-Buffering for Large Geometric Datasets	720
<i>Xiaoxu Han, James Cremer, Zhihong Wang, and Jun Ni</i>	
Image-Based Walkthrough over Internet on Mobile Devices	728
<i>Yu Lei, Zhongding Jiang, Deren Chen, and Hujun Bao</i>	
Service-Oriented RunTime Infrastructure on Grid	736
<i>Tae-Dong Lee, Seung-Hun Yoo, and Chang-Sung Jeong</i>	
Quality-of-Service Driven Visual Scheduling in Grid Computing	744
<i>Changqin Huang, Guanghua Song, and Yao Zheng</i>	
AFEC: An Advanced FEC Algorithm for Video Transmission Control over the Grid	753
<i>Yubo Tan, Yuxing Peng, Sikun Li, and Fujie Chen</i>	
Delayed State Consistency in Distributed Virtual Environments	761
<i>Yuxing Peng, Yongjun Zhang, and Sikun Li</i>	

XVIII Table of Contents

Visual Semantic Query Construction in Dart Database Grid 768
 Yuxin Mao, Zhaohui Wu, and Huajun Chen

A Distributed Data Server in Grid Environment 775
 Bin Chen, Nong Xiao, and Bo Liu

Author Index 783

ART Based Predictive Caching System for XML P2P Database

Wan-Song Zhang and Da-Xin Liu

Department of Computer Science and Technology, Harbin Engineering University, China
zhangwansong@hrbeu.edu.cn

Abstract. Caching and prefetching are well known strategies for improving the performance of XML(eXtensible Markup Language) database systems. When combined with query results clustering, these strategies can decide to cache and prefetch XML documents with higher accuracy. In this paper, we present a predictive system for caching XML P2P database. Our method for clustering XML query results is based on the ART1 neural nets. We compare the quality of caching replacement strategy basing on ART1 with that of the LRU caching strategy. The results of our study show that our method can improve the caching performance of XML P2P database.

1 Introduction

With the growing popularity of Internet, there is an increasing amount of information being distributed and shared in XML format. The increasing frequency of transactions between distributed requirements produces a huge amount of XML documents. The peer-to-peer (P2P) computational model has been emerged with many applications. An XML P2P database is built as the container of a vast amount of XML documents for sharing data, computational resources, etc. Therefore, the XML P2P database acts as the repositories for storage, management and the query-answering interface in open-ended and dynamic networks.

Due to the growing demand by web applications for retrieving information from multiple remote XML sources, it becomes more critical to improve the efficiency of current XML query engines by exploiting caching technology to reduce the response latency caused by data transmission over the Internet. Inspired by the perfecting caching idea [5,6,10,14,15], which utilizes cached queries and their results to answer subsequent queries by reasoning about the frequent sub-queries, we propose to build such a caching system to facilitate XML query processing in the XML P2P database.

Li Chen [8] proposed a semantic method to deal with the page replacement in context with Web environment. One major difference between semantic caching systems [4,9,10,11], and the traditional tuple [5,7,12] or page-based [3] caching systems is that the data cached at the client side of the former is logically organized by queries, instead of physical tuple identifications or page numbers. To achieve effective cache management, the access and management of the cached data in a semantic caching system is typically at the level of query descriptions.

The rest of this paper is organized as follows. In section 3, the ART-Cache system architecture is described firstly. In order to using ART method, the vector mapping procedure is showed secondly. Finally, the ART-Clustering algorithm is described in detail. In section 4, the experiment is studied. In section 5, the conclusion is described.

2 Motivation

Maintaining a cache of XML P2P database can dramatically reduce demand on the network as well as latency seen by the user. The replacement policy for a cache determines which data to cut to make room for new data to be brought into the cache. An important responsibility of cache management is to determine which data items should be retained in the cache and which ones should be replaced to make free space for new data, given limited cache space.

Motivated by the described above, we proposed ART1 neural networks based clustering method to mining the predictive rules for the cache replacement. The ART1[1] is a modified version of ART[2] for clustering binary vectors. The advantage of using the ART1 algorithm to cluster XML query patterns is that it adapts to the change in user's access patterns over time without losing information about their previous access patterns. Furthermore, our research is complementary to previous caching approaches, and deals with a different form of caching: caching of XML queries result trees in an XML P2P database main memory, so that they can be sent faster to clients that request them.

According to the problem described above, the predictive caching problem can be summarized as follows:

How to use an adequate method to cluster the queries to discover the user's query rules from the query logs, that is preload into cache memory the results of the query the user is most likely to ask based on the current user query and the discovered interesting rules. In this paper, ART neural networks are adopted and trained to cluster the queries for generation of the predictive policy rules, which can help XML P2P database cache reduce the response time and the cost of transaction.

3 ART-Cache Predictive System

3.1 System Architecture

In this section, we present a predictive scheme in which we use ART1 based clustering algorithm to cluster users access patterns. The architecture of ART-Cache is shown in Figure 1.

This system is composed by three components. The first component is mining component by ART-Clustering method, which will generate predictive rules. In addition, the ART-Clustering gathers and analysis the XML query logs and generate predictive rules. The second is the answering component, which will answer the result to user. In addition, the third component is built for caching, which will determine how to find the result and the page replacement strategy according to the predictive rules.

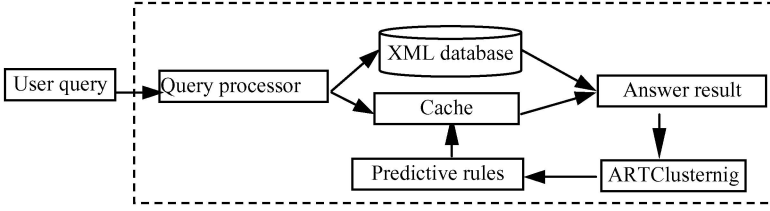


Fig. 1. A Peer with ART-Cache

When a user issues a query, the query processor will judge whether the result can be find in the cache or not. If the result can be find in the cache, the query processor will get the page, in which the result can be composed to send to the user. In this procedure, the query processor will refer the predictive rules generated by ART-Clustering according previous XML query patterns.

3.2 Preparing Query Result Vectors

In fact, an XML query result is a subset of the XML data stored in the XML P2P database. Usually, it is materialized and delivered to user. In addition, it is known that the response time of getting data from cache is faster than directly retrieving from disk. Therefore, we cluster all the query patterns and find the most useful patterns resided in the cache for predicting next query to reduce the response time instead of retrieving from disk again.

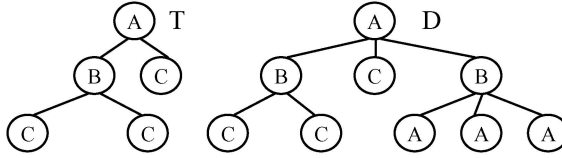


Fig. 2. A query result tree T and a data tree D

XML queries results can be modeled as trees, query result trees, and the query result is a materialized view composed by a set of simple paths. Furthermore, different user focus on different data in a multi-user environment while different user's query logs can be record. Here, the log file has the format: <UID, time, QRT>, where UID is the user identifier and QRT is query result tree which will be defined followed. In order to get the vector of XML queries issued by a certain user, we firstly give the definition query result tree, and then prepare the input vectors used ART neural networks.

Definition 1 QRT(Query Result Tree). A query result tree is a rooted tree $QRT=(V,E,r,label)$. Where V is the vertex set, E is the edge set. The root of the result tree r is denoted by $root(QRT)$.

Given an XML database $D=(D_1,...,D_i)$, D_j is an XML document. A query result tree is a logic tree in response to the answering result issued by a user. Without lost generalization, we assume that a query result tree deprived from one tree.

Definition 2. Let $T = (V_T, E_T, r_T, label_T)$ and $D = (V_D, E_D, r_D, label_D)$ be XML trees. We call T and D the query result tree and the data tree, respectively. Then, we say that the query result tree T occurs in the data tree D if there is a mapping $\varphi: V_T \rightarrow V_D$ satisfying the follows for every $x, y \in V_T$:

1. φ is one-to-one, i.e., $x \neq y$ implies $\varphi(x) \neq \varphi(y)$.
2. φ preserves the parent relation, i.e., $(x, y) \in E_T$ iff $(\varphi(x), \varphi(y)) \in E_D$.
3. φ preserves the labels, i.e., $label_T(x) = label_D(\varphi(x))$.

Then, the mapping φ is called a matching from T into D showed in figure 2.

In ART-Cacher, the UID number is 50 and the frequent QRT number is 100, that is there can be 50 different user issued queries, in which ART cluster find the most 100 frequent queries for experiment.

For each user U , we form a binary pattern vector P_h . For each element P_i in pattern vector P_h , $1 \leq i \leq 100$, if QRT_i issued by the user matched D_j 2 or more times, $P_i=1$, otherwise, $P_i=0$. The pattern vector P_h is the input vector to the ART1 clustering algorithm.

3.3 ART Based Clustering Algorithm

Adaptive Resonance Theory (ART) is a subset in the category of self-organizing neural network, which performs unsupervised batch clustering of input data. Given a set of input patterns, an ART network will attempt to separate the data into clusters.

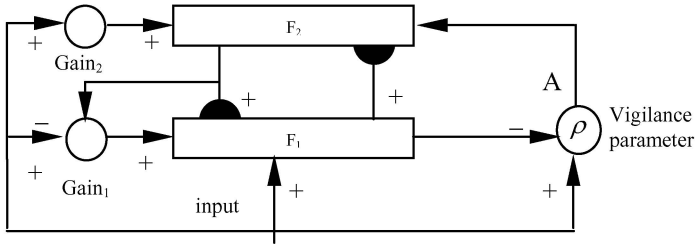


Fig. 3. Architecture of ART1

The dynamics of ART networks consists of the interaction between two layers of processing elements (nodes) in the form of an iterative feedback loop. The first layer in an ART network, termed F1, functions as the short-term memory (STM) for the network. The second layer is termed F2, which is an adaptive layer. The weights between F1 and F2 act as the long-term memory (LTM) for the network. Each node in the F2 layer is a cluster in the set of input patterns and contains the node prototype representing the center of the cluster. The number of nodes in the F2 layer grows dynamically as required to cover the input patterns. For the features of XML data, ART1 is employed in our project.

In ART-Clustering algorithm, each cluster of users is represented by a prototype vector that is generalized representation of XML query patterns frequently accessed by all the members of that cluster.

The procedure for clustering XML query result patterns using the ART1 algorithm describes as follows. The inputs of procedure QRT_Clustering_ART are feature vectors and the vigilance parameter value (δ), while the outputs are clusters of QRTs grouped according to the similarity determined by δ .

Firstly, values are assigned to the control gains Gain_1 and Gain_2 in figure3.

$$G_1 = \begin{cases} 1 & \text{if input } V_{\text{QRT}} \neq 0 \text{ and output from } F_2 \text{ layer} = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$G_2 = \begin{cases} 1 & \text{if input } V_{\text{QRT}} \neq 0 \text{ and output from } F_2 \text{ layer} = 0 \\ 0 & \text{otherwise} \end{cases}$$

The other steps of algorithm QRT_Clustering_ART for clustering XML query results are described as follows:

- (1) Initialization step: Set nodes in F_1 layer and F_2 layer to zero; Initialize top-down

(t_{ji}) and bottom-up (b_{ij}) weights. $t_{ji} = 1$ and $b_{ij} = \frac{1}{n+1}$, where n is the size of the input vector.

- (2) Repeat step 3-10 until all input vectors are presented to the F_1 layer.

- (3) Present randomly chosen input vector $V_{\text{QRT}} = V_1, V_2, \dots, V_i, \dots$ where $V_i = 0$ or 1 at F_1 .

- (4) Compute input y_i for each node in F_2 layer, where $y_j = \sum_{i=1}^{\text{number of nodes in } F_2} V_i \times b_{ij}$

- (5) Determine k , the node in F_2 that has the largest y_k

$$y_k = \sum_{j=1}^{\text{number of nodes in } F_2} \max(y_j)$$

- (6) Compute activation $X_k^* = (X_1^*, X_2^*, \dots, X_{i=200}^*)$ for the node k in F_1 , where

$$X_i^* = t_{ki} \times P_i, \quad i = 1 \dots 100$$

- (7) Calculate the similarity between X_k^* and input P_H using: $\frac{\|X_k^*\|}{\|P_H\|} = \frac{\sum_{i=1}^{100} X_i^*}{\sum_{i=1}^{100} P_i}$

- (8) Compute the similarity calculated in Step 7 with the vigilance parameter:

$$\text{If } \left(\frac{\|X_k^*\|}{\|P_H\|} \right) > \delta$$

```

Begin
  Associate input  $P_H$  with node  $k$ 
  Temporarily disable node  $k$  by setting its activation to 0
  Update top-down weights of node  $k$ 
   $t_{ki}(new) = t_{ki} \times P_i$  where  $i = 1 \dots 100$ 
end
else
(9) Create a new node in  $F_2$  layer
  Begin
    Create a new node  $m$ 
    Initialize the top-down weights  $t_{mi}$  to the current input pattern
    Initialize bottom-up weights for the new node  $m$ 
    
$$b_{im}(new) = \frac{X_i^*}{0.5 + \sum_{i=1}^{100} X_i^*} \text{ where } i=1 \dots 100$$

  end
(10) Goto Step 2.
(11) End

```

The result of QRT_Clustering_ART algorithm are set of clusters, in which XML simple path is as a atom item for clustering. Given these clusters, a query issued by a user can be executed with predictive manner, which is the result retrieve from disk to cache not only including the result but also including the latent query result according to ART clusters.

Supposed, an XML simple path p is a query result issued by a user. If p is a member of cluster C_k , then select the objects in the same C_k with p w.r.t a given radius of C_k . if cache has free space, then directly retrieve XML data from XML P2P database to cache else call the cache replacement policy to determine which object should be replaced. Compared with the LRU replacement strategy, our method in this paper show that the data in cache not only has semantic meaning but also has relations between a sequences of queries issued by a certain user, which affected the performance of cache management.

4 Experimental Studies

In this section, we studied the performance of the proposed method. We worked on PIII 800 PC workstation with 128 Mbytes of memory and 15Gbytes of disk storage. Our experimental studies serve two main purposes. The first purposes is valid the relation between the cache size and hit-radio when the cache size varies. The second purpose is to determine whether or not can help to improve query performance when we change the support number of ART clustering procedure. For this, the hit-radio is that how many simple path queries can be retrieved in the cache instead of finding

from the back-end XML P2P database, which is denoted by δ_h . The main step for predictive caching is the cluster method based ART1 neural nets. The vigilance parameter ρ in ART1 affects the number of clusters. For example, the number of clusters is 15 when the ρ is 0.2 and the number of clusters is 70 when the ρ is 0.7. In this experiment, we adopted that ρ is 0.5 to generate 50 clusters.

For the first experiment, we examine whether the hit-radio and execute time are varied or not when the cache size is changed. Therefore, we set the page range is from 50 to 200.

Depicted by figure 4(a), we can find that the system hit-radio is improved with the increasing system cache size. Compared with LRU method, the ART-Cache has the hit-radio more about 5% than LRU replace strategy, which is the predicted rules generated by ART-Clustering method can improve the performance of XML P2P database cache. This experiment showed only the relation between the hit-radio and cache size. The execute time, however, is another important element which will be affected by the result of ART-Clustering method. Therefore, we set up another experiment to examine the relation between the execute time and minimal support of ART-Clustering.

With the minimal support increasing, the predictive rules is become more and more expert than before, that is the cache become predictive and prefetch pages basing on the clustering result by ART-Clustering algorithm. In figure 4(b), we can find that the execute time varied obviously at minimal support 0.6%.

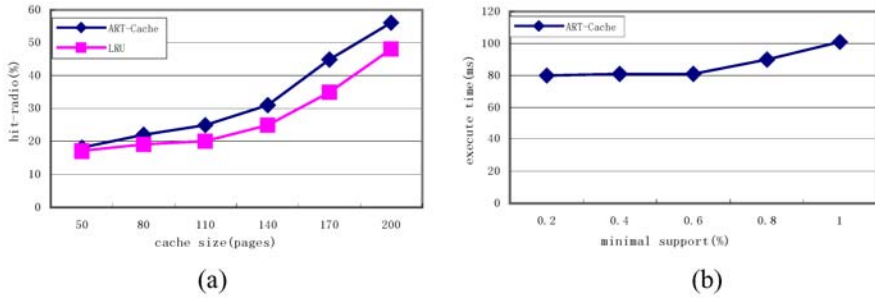


Fig. 4. (a) the relation between cache size and hit ratio; (b) the relation between execute time and minimal support

These experiments show that our ART-Clustering algorithm can generate predictive rules for cache page replacement in most of the tested scenarios. In particular, the ART-Cache replacement strategy will perform better when the minimal support at point of 0.6%.

5 Conclusions

Caching and prefetching technique are useful for XML P2P database when bandwidth is limited. Unfortunately, the increasing spread of XML data seriously hampers

traditional caching techniques for the tree structure model and not rigid characteristics. In order to improve the performance of the cache for XML queries, we presented a system for the efficient caching of XML queries on XML P2P database, where the cache is supported by the predictive rules generated by algorithm QRT_Clustering_ART based on ART1 neural nets. We then perform an experimental investigation comparing our method traditional LRU strategy. The results of our study show that our method can improve the caching performance. In general, our scheme can be used efficiently in XML P2P database.

References

1. Carpenter, Gail A., Grossberg, Stephen. Invariant pattern recognition and recall by an attentive self-organizing ART architecture in a nonstationary world. In Maureen Caudill and Charles Butler, editors, Proceedings of the IEEE First International Conference on Neural Networks, San Diego, CA, pp. II-737-II-745, June 1987, IEEE
2. Heins, Lucien G., Tauritz, and, Daniel R. Adaptive Resonance Theory (ART): An Introduction. Internal Report 95-35, Department of Computer Science, Leiden University, pages 174-185, The Netherlands, 1995
3. Phoha V. V., Iyengar S.S., and Kannan R. Faster Web Page Allocation with Neural Networks. IEEE Internet Computing, Vol. 6, No. 6, pp. 18-26, December 2002
4. Tian W., Choi B., and Phoha V. An Adaptive Web Cache Access Predictor Using Neural Network. In Proceedings of the 15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. pages 450-459, IEA/AIE, Cairns, Australia, June 2002
5. A. Nanopoulos, D. Katsaros, Y. Manolopoulos. A data mining algorithm for generalized web prefetching. IEEE Transactions on Knowledge and Data Engineering, 2002
6. A.Nanopoulos,D.Katsaros. Exploiting Web Log Mining for Web Cache Enhancement. WEBKDD 2001
7. J. Challenger, A. Iyengar, and P. Dantzic. A Scalable System for Consistently Caching Dynamic Data. Proceedings of IEEE INFOCOM '99
8. Li Chen, Elke A. Rundensteiner, Song Wang. XCache: a semantic caching system for XML queries. SIGMOD Conference 2002
9. Liang Huai Yang, Mong Li Lee, Wynne Hsu. Efficient Mining of XML Query Patterns for Caching. VLDB, Berlin, Germany, September 2003
10. Greg Barish and Katia Obraczka. World Wide Web Caching: Trends and Techniques. IEEE Communications Magazine Internet Technology Series, May 2000
11. D Guillaume and F Murtagh. Clustering of XML Documents. Computer Physics Comm., vol. 127, pp. 215-227, 2000
12. A. Nierman and H.V. Jagadish. Evaluating Structural Similarity in XML Documents. Proc. Fifth Int'l Workshop Web and Databases, June 2002
13. J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. DeWitt, and J. Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. Proc. 25th Int'l Conf. Very Large Data Bases, pp. 302-314, 1999
14. LIU Rui-Hong, CAO Dong-Qi. Improving The Performance of Web Information Caching Management By Adaptive Mechanism. Journal of Computer Research and Development, pp.589-594, 2000
15. Hosseini-Khayat. On optimal replacement of nonuniform cache objects. IEEE Transactions on Computers, 49(8), 769-778, 2000

DART-FAS: Federated Access Service on Database Grid*

Guozhou Zheng, Zhaohui Wu, and Chang Huang

College of Computer Science, Zhejiang University, Hang Zhou, 310027, China
{zzzg, wzh, changhuang}@zju.edu.cn

Abstract. The emergence of a service-oriented view of computation and data resources on the grid raises the question as to how database resources can best be deployed or adapted for use and management in Grid. Several proposals have been made for the development of Grid-enabled database services. However, there are few service orchestration frameworks for constructing sophisticated higher-level services that allow database dynamic federation and distributed transaction to take place within a Database Virtual Organization. As the phase II of DartGrid we propose DART-FAS, a service orchestration focusing on the construction of dynamic federation and federated access service in Database Grid. This paper will discuss the architecture, core components and primary processes of DART-FAS.

1 Introduction

The emergence of a service-oriented view of computation and data resources on the grid [1] raises the question as to how database resources can best be deployed or adapted for use and management in such an environment. Several proposals have been made for the development of Grid-enabled database service. The Spitfire [3] service grid-enables a wide range of relational database systems by introducing a uniform service interface, data model, and network protocol and security model. Ongoing work in the Database Access and Integration Services Working Group of the Global Grid Forum [DAIS-WG, <https://forge.gridforum.org/projects/dais-wg>][5][6] is developing a proposal for a standard service-based interface to relational and XML databases in the OGSA setting. DAIS-WG provides a specification for a collection of generic grid data access and data transport interfaces [4]. OGSA-DAI [7] implement the specification of DAIS-WG and provide several basic services for accessing and manipulating data in Grid, including DAISGR (registry) for discovery, GDSF (factory) to represent a data resource and GDS (data service) to access a data resource. OGSA-DQP [8] is a proof of concept implementation of a service-based distributed query processor on the grid, which provides Grid Distributed Query Service (GDQS) for query process and Grid Query Evaluation Service (GDES) for query evaluation.

* This work is supported in part by China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid: Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

OGSA-DQP implements a service orchestration framework both in terms of the way its internal architecture handles the construction and execution of distributed query plans and in terms of being able to query over data and analysis resources made available as services [8]. However, there are few service orchestration frameworks for constructing sophisticated higher-level services that allow database dynamic federation, federated access, federated management and distributed transaction to take place within a Database Virtual Organization (VO) [2]. In phase I of our DartGrid [9] project, we proposed the Database Grid named DART [10], a framework that wraps the existent relational database systems and expose a series of functional services at different levels in support of database resource management in the Grid context. Now as the phase II of DartGrid we propose DART-FAS, which focus on the construction of dynamic federation and federated access service in Database Grid. We give the architecture of dynamic federated construction and federated access service, and introduce core components and primary processes of DART-FAS.

This document is structured as follows. Section 2 gives an overview of DART and DART-FAS, also describes the architecture and function of DART-FAS. Section 3 introduces core services and components in DART-FAS. Section 4 indicates how these services and components can be used to construct dynamic federation and provide access to federated database resources by primary processes in DART-FAS. Section 5 identifies some issues relating to distributed transactions control in DART-FAS. Section 6 presents some conclusions and future work.

2 Overview

DART-FAS are one of high level services in the layered architecture framework of DART, as shown in Figure 1.

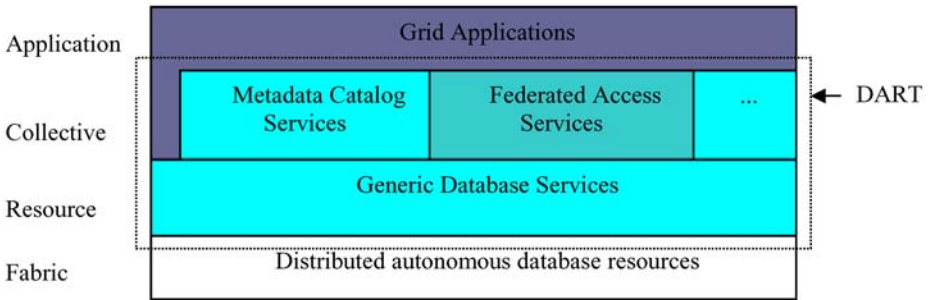


Fig. 1. Layered Architecture of DART

There are four layers in architecture of DART:

- **Fabric layer:** The DART Fabric layer provides the distributed autonomous database resources to which shared access is mediated by Grid protocols; include relation database, OO database and XML database. A database resource is the basic data sharing entity.

- **Resource layer:** The DART resource layer provides generic database services such as metadata publish, database statement, data delivery, transaction control and basic grid service PortType including Factory and Notification.
- **Collective layer:** The DART Collective layer provides services orchestration includes resources metadata catalog, distributed query process, the construction of dynamic federation and federated access service.
- **Application layer:** The DART Application layer provides grid applications on DART.

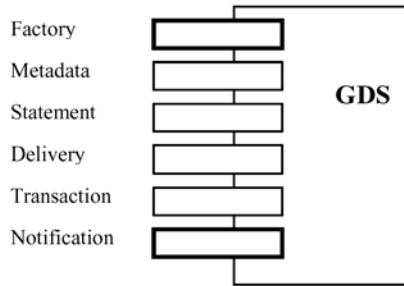


Fig. 2. Grid Database Service portTypes

Grid Database Service (GDS) is the generic database service wrapped to distributed autonomous database resources and exposes a series of functional service portTypes for database, as shown in Figure 2, including:

- **Metadata.** The metadata portType provides access to metadata about the DBS and publishes a description of a service to a service registry such as metadata catalog service.
- **Statement.** The statement portType allows queries, updates, loads or schema change operations to be sent to a database system for execution.
- **Delivery.** The delivery portType is a means by which potentially large amounts of structured data is moved from one location to one or more others. The delivery mechanism should be considered complementary to protocols such as GridFTP.
- **Transaction.** Transactions are crucial to database management, in that they are central to reliability and concurrency control.
- **Factory.** The factory portType must be included when define a Grid Service and create new transient service instance.
- **Notification.** The notification portType is also a OGSA interface, that allows clients to register interest in being notified of particular messages and supports asynchronous, one-way delivery of such notifications.

DART-FAS provide collective construction and access of federated database resource in context of Grid, which build on these generic database service. The prime functions of DART-FAS are:

- **Federated Management.** Federated management services could be envisaged specifically for creating, administering, monitoring, and maintaining federation within a Grid setting. As a dynamic loosely couple system, DART-FAS focus on

schema import and mapping from component DBS to federation, provides referential integrity between component DBS and federation.

- **Federated Database Statement.** Database statements allow queries, updates, loads or schema change operations to be sent to federation for execution. Statements on DART-FAS provide heterogeneity, distribution, and location transparency and produce a virtual database to which the application interfaces.
- **Distributed Transaction.** Transactions are not database-specific artifacts – other programs (e.g. OMG) can and do provide transactional features through middleware services that conform to industry standards. But Grid operations should typically be optimize execution for high-volume, longer duration transactions and conversations.

3 Components

There are four components which make up of a Grid-enable federated database system: Metadata Catalog, support schema export of component DBS and service description of GDS, Schema Manager, support federated schema integration and dynamic schema notification, Query Engine, support distributed query process and federated access, and Transaction Controller, support distributed transaction control, as shown in Fig. 3.

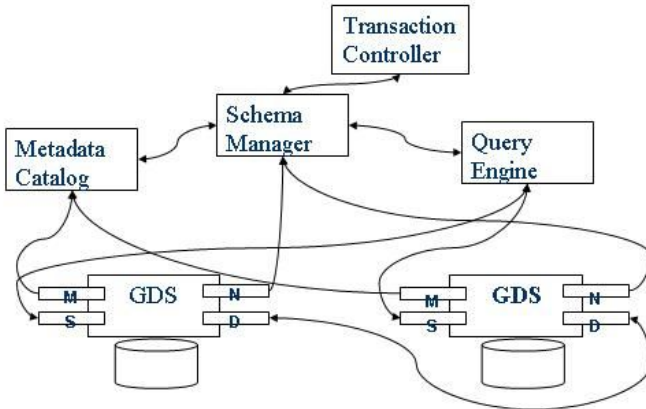


Fig. 3. Architecture and core components of DART-FAS

3.1 Metadata Catalog

The OGSA includes a standard, but abstract, discovery interface that all grid services should support. This existing interface provides the operation that can be used to obtain information about a database service. Metadata Catalog provide database metadata that it could be useful to have access to includes:

- Schema definition: the structure information of accessible tables or views;
- DBMS description: such as the vendor name and version ID.
- Service attributes: physical parameters relevant to GDAMS service.

- **Privilege information:** grantable access privileges for anonymous users. There are two kinds of privileges: system-level privileges (e.g. create table) and table level privilege (e.g. select, and insert).
- **Statistics:** dynamic system attributes, such as CPU utilization, available storage space, active session number and so on. These metadata items can be used for system performance evaluation and resource selection.

3.2 Schema Manager

Schema Manager is the most important component of DART-FAS, which is the start point that setups up the Grid-enable Federation. It is responsible for obtaining meta-data information of Component DBS participating the federation and be shared by the data owner from the Metadata Catalog. It also guarantees the Quality of Service for GDS in the context of open, dynamic grid environment. But, Schema Manager does not focus on complicated conversion and transformation of data model or the data integration by automatic matches. It provides the management interface to integrator, which maintenance the mapping or converting setup by integrator. It also exports the federated data schema to Query Engine for doing Database Statements, at the same time, and to Transaction Controller for providing access control according to the virtual database and supporting lock mechanism on view of federation.

3.3 Query Engine

The role of Query Engine is to allow individual queries or updates to access multiple databases, thereby allowing the system to take responsibility for query optimization and efficient evaluation. When a query that joins data from tables in different databases of Database Grid is submitted to the Query Engine, it will be parsed and optimized, to produce an execution plan by obtaining schema information of relevant databases for joining from the Metadata Catalog. Then when the results of the sub queries are collected and joined by the Query Engine, It can be delivery by long-running asynchronous operations or an opportunity for redirection that is considered complementary to protocols such as GridFTP.

3.4 Transaction Controller

The Transaction Controller maintains properties of the transaction into which the query or update falls. These properties include consistency levels, save point information, distributed transaction state, etc. The issues of distributed transactions will discuss detailed in Section 5.

4 Processes

A DART-FAS process is divided into 3 phases as shown in figure 4, 5, 6.

• Phase 1: Resources discovery and federation construction

At the beginning of DART-FAS, the client browses all the database resources in Database Grid from Metadata Catalog, then sends a “CreateFederation” operation

message with GSHs of GDS to Schema Manager through the Factory PortType, which is the basic port type of a Grid Service. Schema Manager obtains participatory tables' schema by using ImportSchema PortType and locating with GSHs.

- **Phase 2: Integration and maintenance of dynamic federation**

Schema Manager does not think of federated schemas integration, which just focuses on maintenance mapping from integrator. It uses SchemaNotification PortType to notify change of schema and Quality of Service.

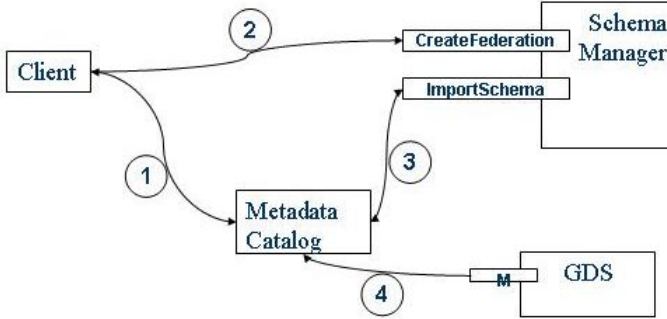


Fig. 4. Phase 1: Construction of loosely couple federation

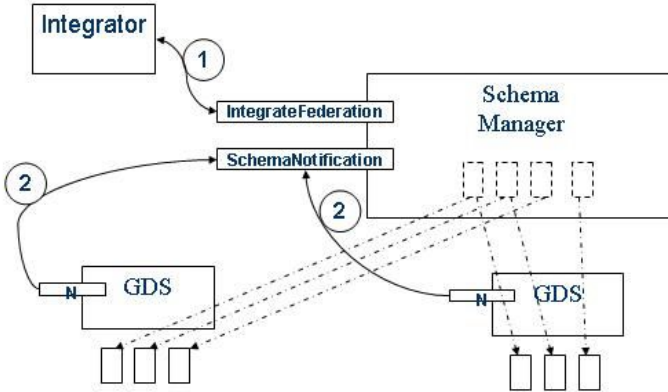


Fig. 5. Phase 2: Integration and federation maintenance

- **Phase 3: Federated access and distributed query**

When phase 1 and 2 is ready, the DART-FAS can accept federated access and distribute query from Grid client or applications. Federated view will be sent through SchemaExport PortType, and federated access statement will be operated by Statement-Dispatch PortType of Query Engine. The result will delivery synchronically or asynchronously straight to the client, or indirect to the third-party.

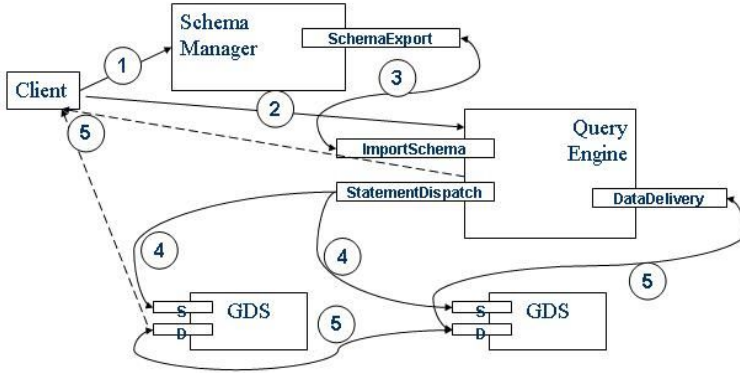


Fig. 6. Phase 3: Federated access and distributed query process

5 Issues for Distributed Transactions

Although transactions are a fundamental concept for database operations the Grid requires additional and more flexible mechanisms for controlling requests and outcomes than are typically offered by traditional distributed and database transaction models. Some specific differences between the Grid and traditional object transaction environments are:

- Multi-site collaborations that often rely on asynchronous messaging.
- Operations across the grid inherently are composed of business processes that span multiple regions of control.
- Grid operations typically optimize execution for high-volume, long duration transactions and conversations.

An incremental approach to distributed transaction on the Grid is suggested:

1. Construction of a core activity service model that provides the capability to specify an operational context for a request or series of requests, controlling the duration of the activity, and defining the participants engaged in the outcome decision.
2. Development of a high level service that provides implementation of patterns typical in a Grid environment, i.e. a two-phase commitment semantic or patterns for compensation, reconciliation, or other styles of collaboration.

6 Conclusion and Future Work

We have introduced a service orchestration framework for constructing sophisticated higher-level services that allow database dynamic federation, federated access, federated management and distributed transaction to take place within a database virtual organization. We have described the grid-enable dynamic federation and federated management architecture for database systems. We focus on several fundamental functionalities: resources discovery and federation construction, integration and main-

tenance of federated view, federated access and distributed query. We also discuss the requirements of distributed transaction in grid setting. We have deployed the middleware on tens of distributed Traditional Chinese Medicine databases, which is part of our TCM Info-Grid program [12], to make performance evaluation and optimization. After that, we plan to design and implement distributed transaction controller and others high-level auxiliary grid services driven by the TCM applications, such as authorization, transformation, replication, and accounting.

References

1. Foster, C.Kesselman, J.Nick, S.Tuecke. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
2. I. Foster, C.Kesselman, S.Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Journal of Super-computer Applications, 15(3), 2001, pp. 200-222.
3. W. H. Bell, D. Bosio, W. Hoschek, P. Kunszt, G. Mc-Cance, and M. Silander. Project Spitfire – Towards Grid Web Service Databases. In *Global Grid Forum 5*, 2002.
4. N.P. Hong, A.Krause, S.Laws, S.Malaika, G.McCance, J.Magowan, N.W.Paton, *Grid Database Service Specification*. DAIS WG, Global Grid Forum, Feb 16, 2003.
5. V. Raman, I.Narang, C.Crone, L.Haas, S.Malaika, T.Mukai, D.Wolfson and C.Baru. *Services for Data Access and Data Processing on Grids*. DAIS WG, Global Grid Forum, Feb 9, 2003.
6. M.P.Atkinson, V. Dialani, L. Guy, I. Narang, NW. Paton, D.Pearson, T.Storey and P.Watson *Grid Database Access and Integration: Requirements and Functionalities*. DAIS WG, Global Grid Forum, Feb 2003.
7. A. Krause, T. Sugden, and A. Borley. Grid Data Service. Technical report, OGSA-DAI, 2003. Document Identifier: OGSA-DAI-USER-UG-GDS-v4.1, July, 2003.
8. OGSA-DQP: A Service-Based Distributed Query Processor for the Grid, *Alpdemir, M.N., Mukherjee, A., Gounaris, A. et.al*. In Proceedings of the Second e-Science All Hands Meeting, Nottingham, UK, 2-4 September 2003 *Cox, S (ed)* 2003.
9. Zhaohui Wu, Chang Huang, Guozhou Zheng. Database Grid: An Internet Oriented Database Resource Management Architecture. In *Proceedings of International Workshop on Grid and Cooperative Computing, (GCC 2002)*. Sanya, China, Dec 2002.
10. Chang Huang, Zhaohui Wu, Xiaojun Wu, Guozhou Zheng. Dart: A Framework for Database Resource Access and Discovery. In *Proceedings of International Workshop on Grid and Cooperative Computing (GCC 2003)*.
11. Zhaohui Wu, Huajun Chen, Chang Huang, Jiefeng xu. Towards a Grid-based Architecture for Traditional Chinese Medicine, In Proceedings of Challenges of Large Applications in Distributed Environments International Workshop on Heterogeneous and Adaptive Computation, June 2003.

A Scalable Information Grid Architecture Based on P2P and Web Service Technologies*

Dugki Min¹ and Eunmi Choi²

¹ School of Computer Science and Engineering, Konkuk University
Hwayang-dong, Kwangjin-gu, Seoul, 133-701, Korea
dkmin@konkuk.ac.kr

² School of Business IT, Kookmin University
Chongnung-dong, Songbuk-gu, Seoul, 136-702, Korea
emchoi@kookmin.ac.kr

Abstract. We propose a Universal Information Sharing and Searching (UISS) system that has a loosely coupled multi-cluster P2P architecture based on Web Service standard technology. It provides a P2P group information sharing and searching services between Special Internet Groups (SIGs), and scalable and fault-tolerant group-sharing P2P Web services with standardized service interfaces and message protocols. We introduce the overall architecture of the UISS system with GISS (Group Information Sharing and Searching) server components. We show three levels of information sharing in the UISS system: a simple case of information sharing within a group, intra-cluster information between tightly collaborated groups, and inter-cluster sharing between loosely cooperated clusters.

1 Introduction

In this paper, we propose a kind of P2P system, called the UISS(Universal Information Sharing and Searching) system. However, compared to ordinary P2P applications, this system follows Web Service standards and is based on Web Service messaging architecture. Thus, we call it ‘P2P over Web Service’ or simply ‘P2P/WS’. According to Gartner Group’s definition, P2P is point-to-point interaction at the edge of the Internet, facilitated by virtual name spaces [1]. According to how to implement discovery mechanism between peers, existing P2P systems can be classified into two types; decentralized P2P and centralized P2P. Decentralized P2P systems do not have any centralized server. Instead, they use their own application-level routing mechanisms for discovering peers. Gnutella, Freenet, OceanStore are the examples of decentralized P2P class [2,3]. On the contrary, centralized P2P systems employ a centralized server to find the appropriate peer that contains desired information or resources. Centralized servers

* This work was supported by the Korea Science and Engineering Foundation (KOSEF) under Grant No. R04-2003-000-10213-0. This work was also supported by research program 2004 of Kookmin University in Korea.

keep and maintain the index DB of shared resources including their metadata and URLs[6]. Napster, Instant Messengers such as AOL and ICQ, and Mojo Nation are classified in this centralized P2P class [4,5]. Once a peer discovers a proper peer, they directly communicate each other according to point-to-point distributed computing protocol. Because of the features of centralized architectural structure and the well-defined standards of Web Service, Web Service technology is used as an underlying infrastructure to construct a centralized P2P system. The UDDI registry takes the role of virtual name spaces to find the location reference of other peers. The WSDL and SOAP are used for standard service descriptions and invocations.

The UISS system is a P2P group information sharing system that provides information sharing and searching services between Special Interest Groups (SIGs) on Internet. The UISS system has a loosely coupled multi-cluster architecture based on the Web Service standard technology. Compared to ordinary personal file sharing P2P systems, the UISS system has more scalable and reliable architecture that consists of multiple grid clusters loosely collaborating in a P2P way based on the Web Service architecture. The UISS follows two standards with regard to content management. In order to handle a content in a standard format, Dublin Core standard[7] is applied. For classifying contents and content servers in a standardized way, the Open Directory schema standard[8] is employed.

This paper is organized as follows. Section 2 introduces the architecture of the UISS system. We describe the system architecture in three points of view: Intra-Group Sharing, Intra-Cluster Sharing, and Inter-Cluster Sharing. A prototypical implementation of the UISS system is presented in Section 3. We summarize in Section 4.

2 The Information Grid P2P Architecture

The UISS information Grid system, presented in this paper, provides clients a single information sharing and searching service whose actual databases are distributed over the Internet. For aggregating a huge number of distributed servers into a single system image, the most important design goal of the UISS system is transparency and scalability of architecture. This section presents the scalable Grid P2P cluster architecture of the UISS system with group-sharing servers in loosely-coupled Web Service clusters.

2.1 Overall Architecture

The overall logical architecture of the UISS system is shown in Figure 1. The essential component of the UISS system is the GISS (Group Information Sharing and Searching) server that is an information storing and sharing server of a SIG (Special Interest Group). The GISS server provides three major functions. First, a GISS server works as a portal to the UISS system for sharing and searching information service. Acting as a service agent, a GISS server provides users an illusion of single system image for the entire UISS Grid system. Next, a GISS

server provides an information storage service to users. Authorized users can store their contents at the server. When storing contents, the user gives its meta information, such as content name, content type, creation data, category, permission, credit and its URL. The content type specifies the multimedia type of the content. The category, called the Content Category, is used to classify the content into categories. The permission indicates its access scope. The content storage of a GISS server is called GIB (Group Information Base) since it is an information base for the group users. The contents of GIB can be accessed only by authorized users, depending on their access property, i.e., permission. Finally, a GISS server maintains a directory of (meta information, URL) pair for all the managed contents whose meta information has known to the GISS server. By means of this directory, called meta-index, a GISS server provides a directory service of the managed contents to users or other GISS servers.

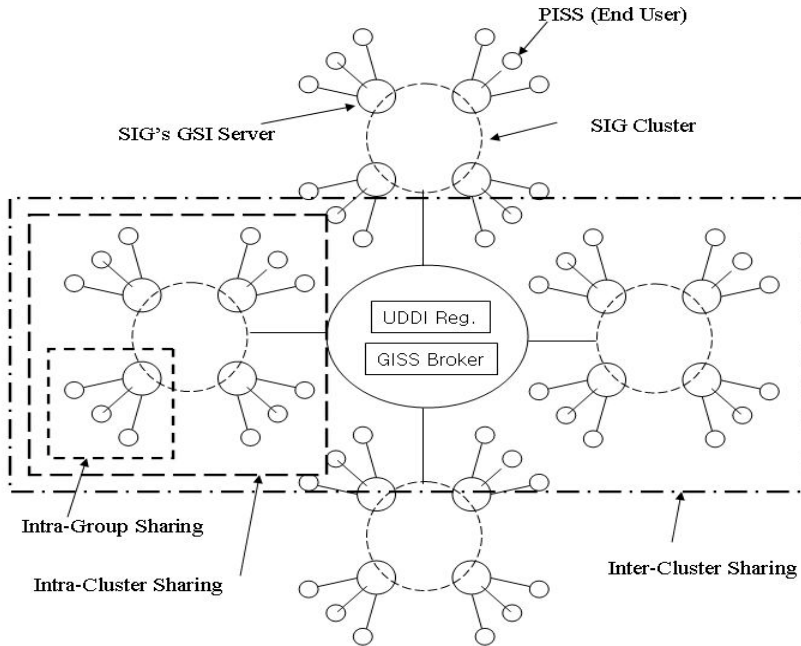


Fig. 1. The Overall Architecture of UISS System

Basically, there are two types of managed contents in a GISS server. One is the content stored in its GIB. The other is the content that is stored at the personal shared storage on directly connected users' PCs or devices. We call this personal shared storage PIB (Personal Information Base). When a user connects to a GISS server, the meta information of PIB is uploaded to the connected GISS server and the meta-index of the GISS server is updated accordingly. That is, a GISS server takes the role of a centralized meta-index server in a centralized

P2P model. Uploading PIB contents to the GISS server connected is an optional. A GISS server maintains the meta-index of its online PIBs as well as its GIB, but not all the GIBs of other GISS servers. When a GISS server needs to search remote contents which locate at other GISS servers, the GISS server contacts the GISS Broker that provides a directory service of all GISS servers.

In order to accelerate searching speed, GISS servers that frequently collaborate can form a cluster, called SIG cluster or GISS cluster. The GISS servers that joined the same SIG cluster, merge and synchronize their meta-indexes in order to construct the global meta-index. Thus, the global meta-index has the index of all the contents that locate at any GISS server of the cluster. Since this global meta-index is replicated to all GISS servers in the cluster, a user can find out a content stored within the cluster by just searching the local global meta-index of the directly connected GISS server, i.e., without contacting other GISS servers through the GISS Broker. This tight clustering mechanism between GISS servers creates replicated global meta-indexes so that the UISS information service becomes more fast, robust and scalable.

In order to understand the UISS architecture in more detail, we explain three cases of information sharing on the UISS system. Section 2.2 gives the simplest sharing within a group. Section 2.3 describes sharing within a cluster where a number of GISS are synchronized. In Section 2.4, we show a general inter-cluster sharing situation where a number of clusters are loosely cooperated.

2.2 Intra-group Sharing

As a simple case, we consider a situation of sharing within a single Special Interest Group (SIG). All members of the group generally share a file server that stores shared group information in it. The centralized group server is the GISS server. As shown in Figure 2, the GISS server contains a number of components to perform operations regarding the group information sharing. The Metadata Manager takes the role of meta-index management. The Metadata Manager is similar to a centralized P2P index server. It maintains metadata of group information shared and stored on the group file storage, i.e., Group Information Base (GIB). The GIB is accessed only through the GIB Manager. The GIB Manager is in charge of managing GIB of the group. When a user activates the PISS, the PISS connects to a GISS server, registers itself to the GISS server, and uploads metadata that describes the user's library. A library is a collection of files that a user is willing to share. The shared library is stored in a specific directory, called PIB. The Metadata Manager also contains the meta-index information of connected PIBs, so as to consider them during searching process. All services of GISS server to PISS are provided by the Service Station in form of Web Methods.

2.3 Intra-cluster Sharing

This subsection considers more complex situations where a number of SIGs want to share their group information. The SIGs having the same interests might want to frequently share their information. For this case, the UISS system provides

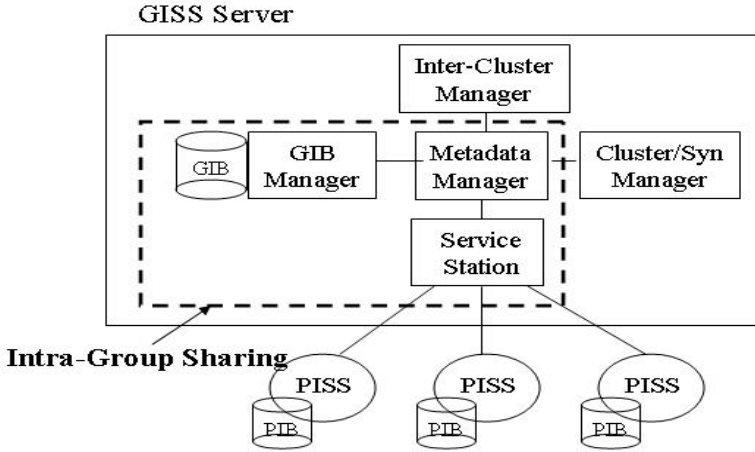


Fig. 2. Intra-Group Sharing

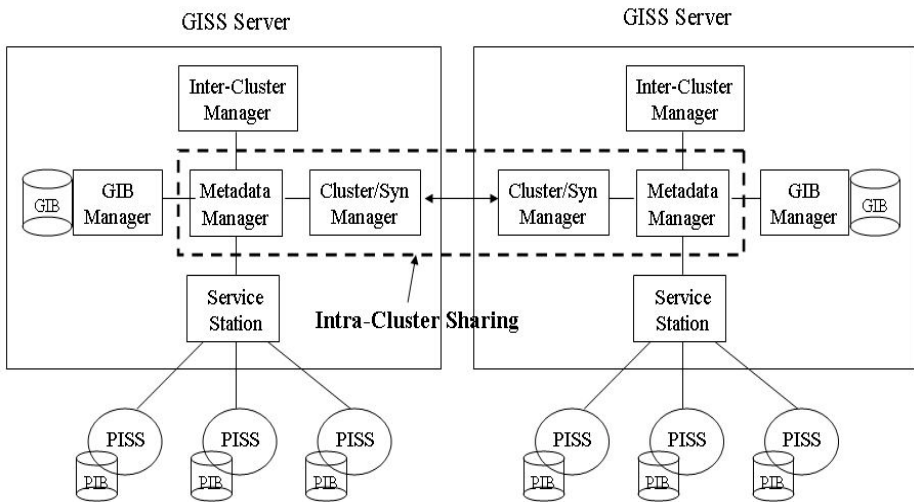


Fig. 3. Inter-Cluster Sharing

the way of GISS clustering, called ‘SIG cluster’. Figure 3 illustrates a SIG cluster with two GISS servers. In each GISS server, there is a component called ‘Cluster/Synchronization Manager’ or simply ‘C/S Manager’. A C/S Manager communicates with other C/S Managers of other GISS servers in the same SIG cluster, and performs meta-index synchronization. Since a Metadata Manager contains the meta-index of its GIB and online PIBs, the meta-index of a GISS server keeps the entire meta-index of all GIBs and the online PIBs in the same SIG cluster after performing synchronization. Therefore, a synchronized GISS

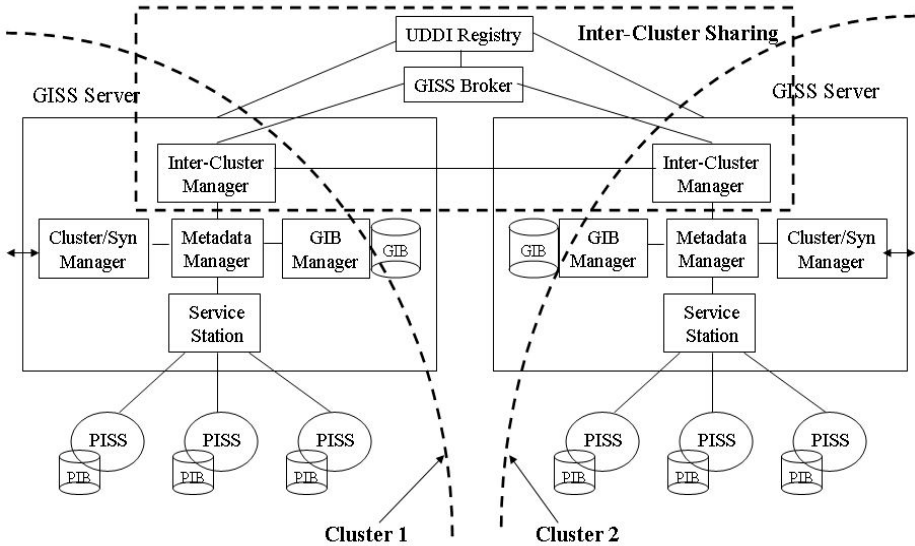


Fig. 4. Inter-Cluster Sharing

server can quickly handle any search query on an item in the entire SIG cluster by itself. The synchronization makes the UISS system fault-tolerant, since there are multiple replicated meta-indexes exist in a cluster.

2.4 Inter-cluster Sharing

In Figure 4, the Inter-Cluster Manager in a GISS server takes the role of inter-cluster sharing. When a GISS server cannot find the matching information of client request within its joined SIG cluster(s), the GISS server starts searching other SIG clusters. The GISS server first contacts the GISS Broker to have the reference of a candidate GISS server that might have the information. The GISS Broker maintains the references and meta data of all GISS servers and SIG clusters. It also periodically monitors the service-level QoS of registered GISS servers and dynamically updates their quality status according to the results of probing. The GISS Broker also synchronizes its directory information to the registry of UDDI. Given a request of a SIG cluster name, it responses the reference of the GISS server that shows the best quality of service in the cluster.

After a GISS server, say GISS A, obtains a reference of another GISS server, say GISS B, from the GISS Broker, the GISS A contacts the GISS B directly and issues a search request of a specific content. The GISS B owns the content, it returns it. If the content is stored somewhere within the cluster, but not in itself, returns the URL of the content. Otherwise, it returns false. The GISS B also provides a number of web methods for exploring interactively its entire meta-index.

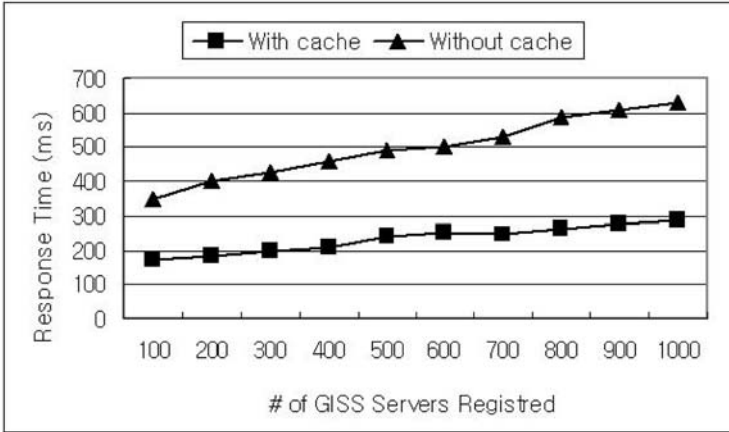


Fig. 5. Performance of Caching

3 Implementation of Prototype System

We developed a prototype of the UISS system in C# on top of Windows .NET environment. A GISS Broker and four pseudo university GISS servers are implemented, and the UDDI registry of Windows 2003 RC1 is employed. As illustrated in the figure, two of the GISS servers are in the same Algorithm category cluster and the rest two are in different categories. Information is shared with an access scope. There are three types of access scope: public, protected, and private. Public contents are open to any GISS server on the Internet. Private contents are closed within the GISS server that owns the contents. Protected contents can be accessed by the GISS servers allied with the owner GISS server.

3.1 Performance Considerations

In order to improve the performance of the UISS system we consider the following performance issues. When a general Web Service operation is invoked, the service object and call objects are created, which results in 1 up to 2 seconds delay. In order to reduce the delay, we develop the Object Pool Manager that generates a number of call objects in advance before needed. In addition, there is another kind of delay in accessing a XML file that is stored in the RMI Registry. In order to fasten accessing time to the XML file in the RMI Registry, we use a DBConnection Pool. These two kinds of simple updates could achieve a better performance improvement from 3 seconds service time up to 0.5 seconds.

There is another consideration in performance improvement. When the number of GISS servers highly increases, requests generated from all GISS servers make the GISS Broker slow down, possibly yielding in a performance bottleneck. To solve this problem, we place a cache in a GISS server in order to speed up information searching time. Once requests are served, the requests are stored in

the cache of the GISS server, so that caching can reduce the number of requests to the GISS Broker. Figure 5 shows performance comparison of response times when there is no cache and when we add a cache in GISS servers. Adding a cache could achieve two times faster response time, compared to the no-cache case.

4 Conclusion

In this paper, we proposed the UISS system that is a loosely coupled multi-cluster P2P architecture based on Web Service standard technology. Compared to ordinary personal file sharing P2P systems, the UISS system has an architecture that consists of multiple grid clusters loosely collaborating in a P2P way based on the Web Service architecture. We developed a prototype of the UISS system in C# on top of Windows .NET environment.

References

1. R. Batcheder: Peer Spaces: The Web Service Desktop. Technology, T-16-2550, Research Node, Gartner Research, Sep. (2002)
2. Beverly Yang and Hector Garcia-Molina: Comparing Hybrid Peer-to-Peer Systems. Proc. 27th International Conference on Very Large Data, Rome, Italy (2000) 561-570
3. B. J. Overeinder, E. Posthumus, F.M.T. Brazier: Integrating Peer-to-Peer Networking and Computing in the AgentScape Framework, Proc. of the Second International Conference on Peer-to-Peer Computing (P2P'02), Sep (2002) 96-103
4. David Barkai: Technologies for sharing and collaborating on the Net. Proc. of the First International Conference on Peer-to-Peer Computing (P2P'01), Aug (2001) 13-28
5. Ozalp Babaoglu, Hein Meling, Alberto Montresor: Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. Technical Report UBLCS-2001-09, Department of Computer Science, University of Bologna, Nov. (2001)
6. Daishi Kato: GISP: Global Information Sharing Protocol - a distributed index for peer-to-peer systems. Proc. of the Second International Conference on Peer-to-Peer Computing (P2P'02), Sep (2002) 65-72
7. Dublin Core Metadata Initiative. <http://dublincore.org>
8. Open Directory Project. <http://dmoz.org/about.html>

Research on Service-Oriented Software Framework

Ying Li, Zhaohui Wu, and Shuiguang Deng

College of Computer Science, Zhejiang University, Hangzhou, 310027, China
zdliying@hzcnc.com, {wzh,dengsg}@zju.edu.cn

Abstract. The service-Oriented development includes two aspects: service design and service integration. Based on the restraint from the correlation among the methods of cooperative components to form web services, we put forward the model of service component, analyze the correlation among the components and discuss the rule to form service; furthermore, propose the concept and method of "invoking in infrastructure" to reduce the dependency relationship during invoking; finally, we study the framework and mechanism to realize asynchronous operations based on existed web service standard and the platform on which service components run.

1 Introduction

Due to flexible linking and highly mobility, Web Services is now increasingly became the artery technology of the integrated distributions and heterogeneous applications[1], and the service-oriented software developing methods and software framework are becoming the new hotspot. First of all, SOAP, WSDL and UDDI have established specifications for the message passing, services definition, discovery and release of WEB SERVICES[2]. All these specifications make the application programs follow a loose coupling, platform-independent model to find each other to interact. Secondly, the maturity of the component based software development (CBSD) technology provides supports for the quick implementation and distribution of WEB SERVICES[3]. Finally, the theory on software architecture provides methodology foundations for the formalized description, comprehension and analyzing the higher level organization of the opening distributed application system[4].

The service-oriented software development focuses on two aspects: service design and service integration. There are already several research in this field: Alan Brown, etc put forward the method to implement software components into software services, and bring forward the object/component/service 3 level developing process[3,4]; WSFL specification[5] tries to keep the business process as kernel to integrate the WEB SERVICES provided by the enterprise virtual organization on the internet, and dynamically form different kind of applications; Z. Maamar studies behavioral theory of compound services by state diagrams[6]; Chris Lüer discusses several key points of runtime environment that supports compound services[7], and Z. Maamar gives out a compound services composition and runtime environment design in detail[8]. Although these researches enriched and deepened the contents of service-oriented software framework, there are still several shortcomings:

Component is the main implement form of WEB SERVICE. The existent study and platform on which component run provide the composition mechanism for collaborative component and the technique that implements component interface as services[3,4,6,7,8], which commonly thought the relationship between methods of components as discrete and independent. However, because component instance is a state machine, and commonly, there is dependent or other kind of relationship among the methods consisting interface and collaborative component methods, which consequently causes the combinational and conformational restriction of the coarse granularity service manipulation which is consisted of thin granularity component methods.

The service invoking defined by SOAP is synchronous in nature, and the interactive model supported by WSDL is only synchronous or irrelevant asynchronous interaction state-independent model. However, not every WEB SERVICE all works by the synchronization; in some situations, the response to WEB SERVICE requirement is not given immediately, but is given at some time after the initial requiring transaction finishes, namely, it needs the asynchronous communication between services. Although WSFL and XLANG can model for the commercial process, and make the application programs seamlessly integrate on the internet without considering its programming language and runtime environment[5], they haven't solved this problem well either. Z.Maamar uses the software Agent to integrate and execute services[8], which could make the Web service invoking asynchronous, whereas Agent haven't become the common information infrastructure, so it is hard to make this practical.

In this paper, we began with the research on the service-oriented software development process, aiming at the problem above, it proposes the service component model and analyses the correlation among the component interface methods and the different modality that components compose into services. Using the method of "invoking point in infrastructure" to reduce call-dependence among services; Finally, based on the existent WEB SERVICE standard and component-running platform to implement asynchronous operations, a fundamental framework is proposed.

2 The Process of Service-Oriented Software Development

The service-oriented software developing process is shown as figure 1:

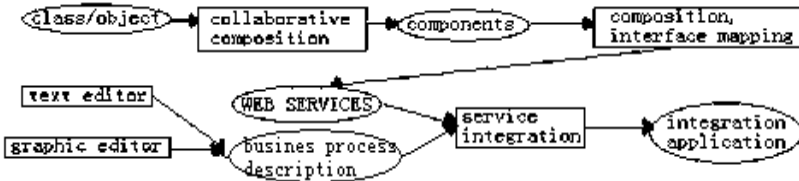


Fig. 1. Service-oriented software developing process

The whole developing process can be divided into four layers: class/object, component(including compounded component and local application), service, integrated

application. Commonly, the process of encapsulating class/object into component is done during design, which is supported sufficiently by the existent component standard (COM, CORBA) and their developing environment. Component-to-service can be refined into two behaviors: component composition and service mapping. Component composition instantiates the member components of compounded component and incorporates the component interface, in order to mask the interaction of components from exterior users and to provide coarse granularity, uniform interface and eliminating session-stating dependency; service mapping describes component interface by WSDL format and makes the exterior user be able to invoke the component interface by SOAP protocol. Component composition can be done not only at the design time, but also at runtime, and the mapping mechanism from component interface to WEB SERVICES is already supported by the component running platform, such as J2EE, . NET and so on. However, these researches and implementations didn't give out behavior guidelines for component composition and interface mapping to services, component interface specification only gives out interface method list, while didn't research on the relationship between interface method, so the designer can only composite components and make the selection and combination of interface operations by experiences and skills. But the fact is that the cooperative and relative relationships among interface methods have a great influence on the implementation of transform component to WEB SERVICES of single instance.

3 Service Design

3.1 The Service Component Model

Component is the undertaker of services, and service is the invoking interface of component, however, not every component can provide services directly. The component that can work as a service provider must be self-inclusive, and relatively independent, which is called service component. At the view point of invoking, a service is a state-independent, single instance compounded component, which manages its internal resource distributing. Component can be compounded, and compounded component includes cooperative relationship among member components. Generally speaking, service component is compounded component, creates internal member component instance, maintains its internal state transforming and permanence, cooperates other member components' relations all by its manage member components, and it makes these behaviors all invisible to the outside world, and provides a unified interface mapping to service for service users to invoke, which is shown as figure 2.

As the components' cooperative relationship in service component incarnates as the relationship of component interface operations, we give out the service component model as below:

Service component :: = (manage component, member component set, member component relation, service)

Manage component :: = (component interface, member component management, service mapping)

Member component management :: = (member component interface integration, member component cooperation)

Member component $:: =$ (component interface, component implementation)

Component interface $:: =$ (interface methods set)

Interface method $:: =$ (interface method identification, parameter list, return value)

Member component relation $:: =$ (all member component interface method set, interface method relation)

Service $:: =$ (message type definition, ports, port binding)

Port $:: =$ (port operation set)

Port operation $:: =$ (service operation identification, input parameter list, output parameter list)

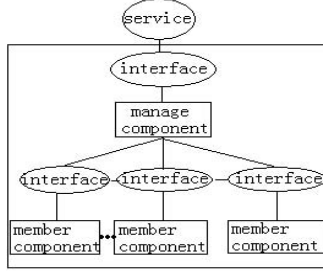


Fig. 2. The service component model

3.2 The Analysis on Relationship of Interface's Methods

In the common component specification, the interface of component only defined the methods which it provided to its external, while didn't define the cooperative relationship among components. The cooperative information of one component with other components is fixed in the implementation of components, so it is hard for component to adapt the environment's changing. The component specification screened too much useful information to its external world, which can't meet the requirement of integrating at interfaces. Our service component specification includes member component relationship, which mainly incarnates on the relationship of its interface methods. The components' interface methods have defined all the information used for internal service components' interaction, and according to these information, it is sufficient to accomplish the mapping process from service component's interfaces to services, without knowing the service components' internal implementation details directly.

As below, there are several kinds of main interface methods' relationship:

1. invoking dependent

Member components in service component provide method sequence $\langle OP, OP_1, OP_2, \dots, OP_n \rangle$, if $OP_1, \dots, OP_n$ are invoked by OP , then OP invoking depends on $OP_2, \dots, OP_n$ directly, and $OP_i (1 \leq i \leq n)$ invokes $OP_i^1, OP_i^2, \dots, OP_i^m$ insides, then OP invoking depends on $OP_i^1, OP_i^2, \dots, OP_i^m$ indirectly. OP and OP_i^m may belong to the same member component, and also may belong to different member components; this kind of invoking dependent relationship can be modeled by sequence diagram in UML. If OP and OP_i belong to different member components, then operation

OP always takes the responsibility of instantiating the member components which OP_i belongs to and maintaining its lifecycle. Looking from the point of view of the service user, commonly it is not necessary to know the existence and invoking implementation details of OP_i , and should also mask the instantiation activity of the member components. Therefore, we design method OP as OP' , which is the manage components' interface method of service components, and do the management of component instance which method OP belongs to, in the implementation of method OP' . According to the service component model, method OP' will be mapping to service operations.

2. state dependent

Service component is an implementation of a state machine; the executing conditions of member components' methods is related with states, and the results of the methods' executing will trigger the switch between states. Consider an ordered sequence $\langle (OP_1, STATE_1, STATE_2), (OP_2, STATE_2, STATE_3), \dots (OP_n, STATE_n, STATE_{n+1}) \rangle$, where $(OP_i, STATE_i, STATE_{i+1})$ stands for method OP_i can execute in state $STATE_i$, and will switch to state $STATE_{i+1}$ after execution, so we say there is state dependent relation in sequence $(OP_1, OP_2, \dots, OP_n)$. Speaking from the nature of service, service components should encapsulate their internal states as much as possible, and provide coarser granularity service operations than member components' methods. Methods sequence $(OP_1, OP_2, \dots, OP_n)$ can be integrated into a method OP that manages components' interfaces, and the parameters of OP is consisted of the union of all parameters of $OP_1, OP_2, \dots, OP_n$, and then remove those parameters who only be passed inside themselves. Looking from the point of view of the service users, service operation OP only has one state transition: $STATE_1 \rightarrow STATE_{n+1}$, if there is $STATE_1 = STATE_{n+1}$, then it can be considered as state-independent.

3. state correlativity

If service components have tuple $((OP_1, STATE_1), (OP_2, STATE_2), \dots (OP_n, STATE_n), STATE_{n+1})$, where $(OP_i, STATE_i)$ stands for method OP_i can be executed in state $STATE_i$ and the whole tuple means that state will switch to $STATE_{n+1}$ after every OP_i executed successfully in their corresponding state. In this situation, we say that there is state correlativity among methods $OP_1, OP_2, \dots, OP_n$. Methods sequence $(OP_1, OP_2, \dots, OP_n)$ can be integrated into a method OP that manages components' interfaces, and the parameters of OP is consisted of the union of all parameters of $OP_1, OP_2, \dots, OP_n$. In fact, we incorporate states $STATE_1, STATE_2, \dots, STATE_n$ into one state.

3.3 The Concept and Method of "Invoking in Infrastructure"

There is inherent relation and dependence for each other among things, so there is cooperation and interaction among service components too. This incarnates in two aspects: cooperative relation and dependent relation. Cooperative relation is self-

inclusive, relatively dependent service components do the exchanges of information flow through the third party's control mechanisms, and then forms a loose coupling cooperative relationship. Dependent relation is one service component's implementation depends on another service component, namely, the implementation details of the service component's service operation invokes the service provided by another service component.

In the implementation of service operations, the invoking of service operations of other service components is rigidly coded in the program. As the client machine's service component, there are two kinds of methods to invoke services: static invoking and dynamic invoking. Static invoking must include the stub code generated by the service description file WSDL which is provided by the service provider in the client machine, in order to solidify service names, ports, operations, binding and the position of service provider in the client machine programs. While dynamic invoking does not need to stub the invoked services, is more flexible, however, it still depends on the service provider. These two all make the implementation and execution of service components tightly coupling to the assigned position of the invoked service components, service names, service port names and operation names, if there is any changes on this information, the invoking service component can't work correctly. The reason lies in the fact that these information is rigidly coded in the programs consequently is compiled into the object code, and the invoking is done by the service operation itself.

Therefore, we give the concept of "invoking in infrastructure". In the implementation of the invoking point of the service components' operations, we only define the parameters to be passed out and the result types to be received and other semantic constrains to be declared, and relegate the real invoking task to the component running platform. The component running platform does the matching selection of service name and operation name according to the definition of operation invoking point, to accomplish the addressing of invoked service and the real invoking task. To the programmers, the component running platform provides the "invoking in infrastructure" programming modeling; when using this kind of programming interface invoking services, programmers needn't to know the position of the service provider, and even needn't to know the service name, port name and operation name, they only need to know the input parameters passing the service and the return data type and other optional information. We implemented this programming interface by extending the J2EE platform, and the design details are shown in section 4.

4 Asynchronous Invoking Mechanism and Running Platform's Infrastructure

WEB service specification and standard does not support asynchronous operations in the obvious style, however, these standards include the infrastructures and mechanisms that can serve as the basis of asynchronous operations. The scheme is to construct the asynchronous actions into the service demander, as discussed in section 3, we know the actual service invoking is implemented by component running platform,

therefore, our method is including the service asynchronous operation mechanisms in the design of component running platform, while differs from the common message passing service mechanisms, such as MMQS, JMS, MQseries, and so on. The service demander sends a requirement as a part of a transaction and keeps on using the executing thread, then another thread processes the responding in a separate transaction. This model can be implemented by the invoking-back mechanism. The nature of design is to regard the exchanging message as data packet, which needn't or doesn't expect any acknowledgement in order to guarantee the transaction to get processed. By using this kind of data packet, and considering the actual asynchronous relation between the two parts, then absolutely detach the message sender from the receiver, the correlator (a correlation or a transaction identification) will associate the responding with their requirements.

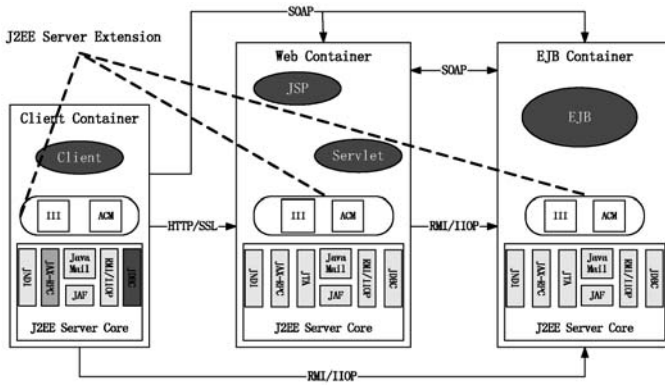


Fig. 3. The extended J2EE platform

In order to implement the "invoking in infrastructure" in the service dependent relations and the service invoking asynchronous mechanism, it needs the service component running platform to provide basic support. Today, the mainly used component platforms are J2EE, CORBA, .NET, and so on. We have extended the J2EE platform to provide this kind of environment, and implemented it on Weblogic, other platforms' design can refer to this. The extended J2EE platform is shown in figure 3.

The III (invoking in infrastructure) is used to support "invoking in infrastructure", ACM (Asynchronism mechanism) is used to provide service asynchronous communication mechanism, all of above are built on the base of J2EE kernel packet. III encapsulates the JAX-RPC in J2EE kernel packet, providing the higher level and more abstract programming interface. Using III, when invoking services, programmers needn't to know the position of the service provider, and even needn't to know the service name, port name and operation name, they only need to know the input parameters passing the service and the return data type and other optional information, the actual invoking tasks are all done by III.

5 Conclusions

Based on the restraint from the correlation among the methods of cooperative components to form web services, this essay studies the correlation of the component interfaces' methods, and proposes the rule to form services, and discusses the different modalities that components integrate to services, thus makes the service operations self-inclusive, relatively dependent action granularity, while not study the integration mechanism and implementation techniques. Putting forward the concept of "invoking in infrastructure" and designing the corresponding method aims to reduce the dependency relationship to invoke. Through invoking-back mechanism and including service asynchronous operation mechanism in the design of component running platform, it can support the service asynchronous invoking and the session stating. We also have designed an infrastructure as the running environment for the method above, which can be implemented on the existent component running platform.

The further work is to study the procedure-oriented new software architecture model, and using this model as the guiding normal form to do the development of WEB SERVICES, the flexible linking of distributed service and the theory and techniques of dynamical integration.

Acknowledgement

This work is supported in part by China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid : Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

References

1. A. Tsalgatidou , T. Pilioura: An Overview of Standards and Related Technology in Web services. Distributed and Parallel Databases, 12(3), Kluwer Academic Publishers, 2002.
2. F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana: Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. IEEE Internet Computing, 6(2), March/April 2002.
3. David Mennie, Bernard Pagurek: An Architecture to Support Dynamic Composition of Service Components. Proceedings of the 5th International Workshop on Component-Oriented Programming (WCOP 2000), Sophia Antipolis, France, June 13, 2000.
4. Alan Brown, Simon Johnston, Kevin Kelly: Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications. A Rational software whitepaper from IBM, April 2003.
5. Frank Leymann: Web Services Flow Language 1.0. , IBM Software Group, May 2001
6. Z. Maamar, B. Benatallah, and W. Mansoor: Service Chart Diagrams - Description & Application. In Proceedings of the Twelfth International World Wide Web. 2000.3

7. Chris Lüer, André van der Hoek: Composition Environments for Deployable Software Components. technical report uci-ics-02-18 – 08/22/2002
8. Z.Maamar, Quan Z. Sheng, Boualem Benatallah: Interleaving Web Services Composition and Execution Using Software Agents and Delegation.
<http://www.cse.unsw.edu.au/~qsheng/papers/maamar.pdf>. 2003
9. Zhang Shi-kun, Zhang Wen-juan, Chang Xin, Wang Li-fu, Yang Fu-qing: Building and Assembling Reusable Components Based on Software Architecture. Journal of Software, VOL.12, NO.9, December 2001

An Accounting and QoS Model for Grid Computing*

Yang Guangwen, Yongwei Wu, Dazheng Huang, and Weimin Zheng

Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China

Abstract. The introduction of economic approaches into grid computing helps resolve some challenging problems in this research area. In this paper, following an account system we present an accounting and QoS model based on T.G., a market-oriented grid computing system with fine-grained parallelism. Our model takes into account some economic factors and also the characteristics of users' behavior so as to build a practical and flexible subsystem whereby T.G. could improve its overall performance and throughput and each participant could benefit himself as much as possible.

1 Introduction

Grid Computing has been a buzzword these years, and certainly we have achieved a great deal of accomplishment. Regretfully thus far we haven't a real open Internet-based system which implements those most original imaginations put forward by our harbingers. Of course some peer-to-peer systems have revealed an enchanting scope for us, but these systems have so long a distance away from a real grid computing. And we do have some grid computing systems running here and there, but they are either the ones based on enterprise intranets that have highly degree of trust and security, or those which is though institution-crossing, but these institutions has highly confidence for one another, or these multiple institutions compose a larger virtual single institution.

It's hard to convict someone that it can be safe and also profitable to let strangers run their suspicious program on his computer or read and write his hard disk. Regarding security, it's such a broad issue that we won't discuss it in this paper, and will focus on the latter one.

Moreover, the value of economic principles has been demonstrated by a lot of practice [1,2,3], for the similarity between the real market and a grid computing system:

- All the actors have full autonomous control over their own resources and services;
- Each actor wish to behave as he like, but everybody also knows that there should be some explicit or implicit regulations to which everybody should be subject at the same time;

* This Work is supported by NSFC (60273007, 60373004, 60373005) and China Post-doctoral Foundation.

- There are common agreements or protocols among actors for which any transaction or deal is to be reached and enforced;
- Each actor has his own different goals and patterns of action;
- Each actor always hopes to benefit himself as much as possible, that is, he always hope to get as much as possible for what he shares for others and to spend as little as possible for each time of use of others' resource;

In brief, to activate a real grid system and make it effective, we should make it clear for every potential participant that he can get something if he contributes something for others, while security to some extent will be assured. Meanwhile the more he affords, the better services he will get. For this scenario to be realized, everybody should have a unique identity to which all of the “bills” is attributed and through which different users can negotiate with each other. Everyone should be reachable and credible.

In this paper, we put forward an account system to tackle the problem of user identity. Based on the account system, a distributed accounting policy is enforced on each machine in our system. Though all the actors are subject to the same policy, there is still enough space for everyone to exert his own ideas. Afterward a QoS model is build based on this policy in order to embody the principle, “the more you pay out, the more you will get”.

2 Related Works

Regarding the practice of economic principles in grid computing, we have already made a simple survey in [4] and here we will focus on the account system, accounting system and QoS model.

An approach of account templates to allocate accounts is presented in [5], the basic idea is to create a pool of “template” accounts that have no permanent association with a user and to create temporary persistent bindings between template accounts and grid users. Each template account uses the regular accounts attributes that are normal for the host system, but the user information refers to a pseudo user, not a real user. This approach is good at addressing the potential “locality” of the grid users’ utilization pattern to support hundred of thousands of grid users. The disadvantage is that the account system is so large that the cost cannot be ignored compared to the whole efficiency of the system.

Condor solves this problem by using one UID (“nobody”) to execute jobs for users that do not have an account in a Condor flock [9]. The PUNCH system uses a similar scheme, where all users are represented by logical user accounts within a single physical account with the ability to use a set of dynamic account bindings for system calls[10]. There are disadvantages to these approaches. If there are multiple jobs on the same system from different users, with all of the users assigned to one UID, it is difficult for the system to distinguish between those users since they all share the same UID.

As for the accounting system, there is little practice and literature to quote. A distributed view of accounting and a methodology for allocating grid resources to computation for use on a grid system are put forward in [6]. In the paper,

such terms as rate, quote, request, chargeable items and so on are addressed or defined. But lots of work is open to do if a practical system wants to be implemented.

When it comes to the QoS model in grid computing, even less can be referred to. And in most cases it is restricted to data grid systems where QoS is mainly utilized to address those issues happening when large magnitude of data is manipulated, as it is presented in [7].

3 Overview of T.G.

T.G.[4] is a Java-centric grid computing system realized based on the campus network of Tsinghua University, Beijing, and you can take the name of T.G. as the abbreviation of “Tsinghua Grid”, but we prefer “Terra and Giga”, for it somewhat reflects the measure of capability and capacity T.G. will aggregate for end users.

- Machine organization and resource model
In T.G., machines are organized in a structure of hybrid tree to reflect the real topology of any WAN and Internet and to make the best of the feature of the locality in networks.
- Programming model and task scheduling
We provide programmers a simple java-based programming mode whereby a task can be built. A integral task is constructed into a tree-like structure. The scheduling process complies with the principle of “enough and no squandering”.
- Security model
Security model of T.G. is based on the java security model and the technology of public-private key pairs, and each machine stores its own security policy in its resource model.
- Monitoring subsystem
Monitoring subsystem tracks the local and the subtree’s information as resource use, network traffic, users’ task execution and so forth.
- The root node and Information Center (IC)
There is a root node when the system is initiated, the machine where this node resides will never shut down. The information center read the data stored in the root node periodically and makes a backup.

4 The Account System

Building an account system might be a challenging problem, as it is addressed in many literatures. However, in T.G. we carry out a simple but effective solution.

As we’ve seen in a lot of commercial sites that maintains a large account system containing hundreds of thousands of accounts, IC also maintains the account system in T.G. and each user has a globally unique account, even though we know the existence of a centralized component might be seen as a disadvantage. Our rational is based on such reasons:

- It is necessary for each actor to occupy a unique permanent account, especially when the two sides of a transaction are strangers to each other. This assures that each actor is identifiable and credible;
- Because the responsibility range of the information center is very limited, its load will be restricted within a reasonable spectrum and won't lead it to collapse;
- Due to the consideration of security, that there is only one institution to exert the management of accounts and identity information is beyond question;

Request of new accounting (request for registration) is issued by one user together with his public key and password through the client software of T.G. toward IC.

It's easy to imagine the probable large quantities of accounts in an Internet-based system. However, each participating machine doesn't have to store all these accounts, even none in fact.

As we've known, the resource and security model needs one specific machine only to preserve the information of some special global accounts and others can be processed as anonymous users in an FTP site. These special accounts might have unordinary access privileges to local resources. In T.G., these special accounts are called VIP.

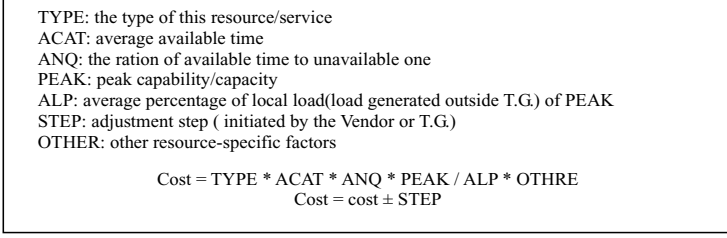
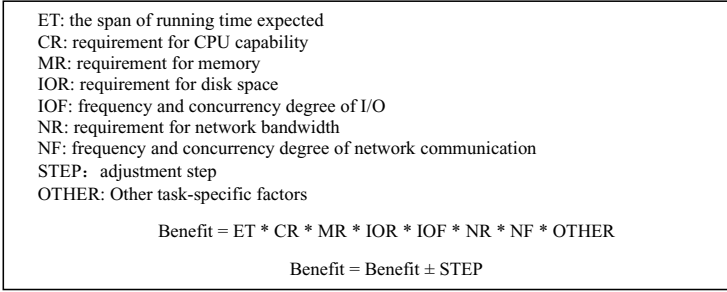
The length of the VIP list is determined by the local administrator and he can also freely remove replace one or more VIP. T.G. recommends a least-use-recently (LSR) algorithm to replace or remove the VIP list. Here we would emphasize again that if he wants there can be no VIP at all and all are anonymous users.

When a task is forwarded here, the runnable byte code should be encrypted with the user's private key. If the user is in the VIP list, the daemon thread (DT) will pick up the user's public key stored locally to verify the code. If the key pairs are not matched, the daemon thread will contact IC to find out whether the user has changed his key pairs and to get his latest public key. If after all these steps are fulfilled and the match is not reached yet, the daemon thread will deny the task; otherwise the task is considered valid.

Our approach has some important differences from other centralized solutions as the project Athena in MIT [8] in that ours implies a map strategy from global accounts to local accounts (either temporary or permanent, up to local administrators).

5 The Accounting Policy in T.G.

We call resource providers vendors and those submitting tasks vendee. The beginning of the accounting process is the cost of a resource and the benefit of a task. And the cost and the benefit will be measured with the same unit, regardless of various types of resources and tasks, partly in order to mask the heterogeneity of different entities in a grid system. Of course for different resources there are different chargeable items, but we can still take features in Figure 1 as common. For any resource there are two costs, one determined by the vendors freely and one by the system, both of which will function in the

**Fig. 1.** How is the Cost Determined**Fig. 2.** How is the Benefit Determined

QoS model. Likewise features in Figure 2 determine the benefit of a task. During a successful matching process when scheduling a task, besides other technical features of a resource, only if the benefit of the task is higher than the cost of the resource, a deal is reached and one time of accounting is started. It should also be noted that the price of this deal is not the cost of the resource but the benefit of the task. So we can also think of the cost of a resource just serves as the threshold: those tasks with benefit lower than it will be denied.

Once a task is started to run, the benefit of it cannot be changed any longer, that is to say the price of this time of accounting will not change, unless some commands from the QoS model demand to do so. Anyway, the QoS model will determine the overall average price of this deal once the task is finished. The price multiplied with the real running time will be the sum the vendee will pay the vendor.

Accounting can also be a complicated process, as it exhibits when a task requiring transmission of large magnitude of data runs. The vendor or the system might add the cost on network traffic. This case is not the same as that when a exclusive communication resource is utilized.

6 The QoS Model of T.G.

While the accounting policy regulates each concrete time of accounting, the QoS model moderates the continuous and dynamic task scheduling and accounting

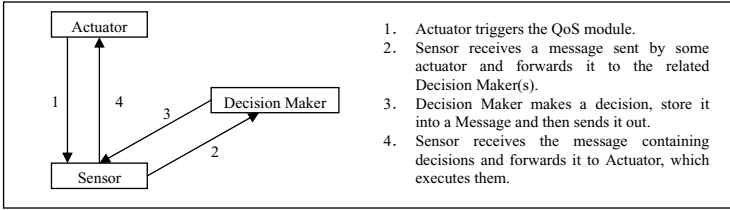


Fig. 3. QoS Module

process with the support of resource management module and monitoring module.

Besides adjusting the cost and benefit, QoS also handles other errands as accounting, adjusting local load level and communication traffic, so factually it serves as both a civil servant and the moderator of the T.G. market.

In the QoS model, there is only one question: how can I benefit myself as much as possible? If I am a vendor, I wish to get as much as possible from vendees and if I am a vendee, I wish to my task would cost me as little as possible in the premise that my task will be executed as I expected. Of course, actors cannot change other factors in the system, other than the cost and benefit, both of which can be adjusted, either by the owner or by the system. And the adjustment by the system is finished forcefully.

The QoS model is made up of three components: sensors, actuators and decision makers, as described in Figure 3. In different scenarios, the QoS model will be activated and functions disparately. Sensors resides on each node in the resource model, each active user has an active Actuator and Decision Maker on behalf of him and there are also an active Actuator and Decision Maker standing for every resource and unfinished request. Each message may have multiple destinations and Sensors on each node decide which local Decision Makers are the potential receivers according to the information in the Message, so T.G. can enjoy the convenience of multicast.

QoS During Data Transmission

For a task of transmitting data, the requirement mainly consists of the bandwidth, whether or not to allow interruption and the benefit, and for a communication resource, the cost is mainly determined by bandwidth, average number of communication flow and average span of available time.

After a deal is reached, the resource starts to transmit data for the vendee. During this period the sensors residing at the two ends of the channel will obtain the current transmission speed periodically and calculate the past average speed (PAS). If the sensors think PAS is too higher or lower than the valued contained in the deal, it will notify the two decision makers on behalf of both the vendor and the vendee.

If the decision maker on behalf of the vendor thinks that it is necessary to increase the bandwidth quota assigned to this deal, it will send a message to

the local sensor, which will forward this message to the local actuator, and the actuator will execute the decision. After this process, if the PAS is increased to the level of the expected speed of the deal, the price won't be changed.

However, if the actuator finds that it is impossible to increase the quota for this decision, it will self-assertively deny the decision from the decision maker and generate a log. Of course, in this case, the eventual price will be lowered than the original value to reflect the actual course.

To protect the behalf of the vendee, the eventual price will never be higher than the benefit of the task, that is, the original price of the deal.

Before the beginning of transmission of data, striking of the bargain means a reservation of bandwidth, despite that the bandwidth might be adjusted later.

The reason for which the decision maker doesn't communicate directly with the actuator, even though they both resides at the same machine, is due to the architecture of T.G., in which there is only one sensor on each node and this sensor only contact with DT. We think this approach reduces both the burden of DT (it doesn't need to receive message from the network) and that of sensors, whose main duty is to receive large numbers of messages from the network and only tell the local DT to which each message should be forwarded.

QoS During Task Execution

The QoS procedure is very similar with that of data transmission, except that here the factor can be adjusted is the priority at which tasks is executed. In some special cases, other tasks might be suspended so as to guarantee the QoS of those tasks with highest priority (also with the highest price).

However, whether the adjustment of the priority of a Java thread will take effect at once relies on the implementation of the JVM and also the operation system, so the eventual price of the deal will be determined accurately only after the task is finished. Still the benefit of the task is the upper limit of the eventual price.

7 Conclusion

To encourage people to share resources, There also exists a basic coefficient, Return Coefficient (RC), in T.G.. If a user shares resource whose objective cost is 80 and the RC is 0.2, he can get a prize with the value of 16 and this prize can be used to compensate the benefit he pays to other for his future tasks. The so-called objective cost is calculated and adjusted only by T.G. in the same way as the Cost, except that all the subjective factors are excluded, so in general the objective cost is less than the Cost.

Plus RC, the accounting system, accounting policy and QoS model make up of the market component of T.G.. As we know, in general there are components called resource brokerage in a grid system, but in T.G, DT also holds the position of brokerage, fulfilling both the resource allocation and deal bargain. It's a concise deployment strategy because reducing the dimension of the whole system has been one of our main objectives.

In addition, our scheduling process cannot be reversed, that is, if a task is forwarded to one node, it cannot be forwarded to this node's ancestors any more regardless of what happens. This algorithm works well when we don't take the cost and benefit into account, but it also implies that for one task, the best economic selection can only be made within a limited scope, not the global system.

References

1. Rajkumar B., David A., Jonathan G., A Case for Economy Grid Architecture for Service-Oriented Grid Computing, In Proc. of 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), In conjunction with IPDPS 2001, (2001)
2. Buyya R, Abramson D, and Giddy J, Economy Driven Resource Management Architecture for Computational Power Grids, In Proc. of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2000), (2000)
3. Cocchi R., Shanker S., Estrin D., and Zhang L., Pricing in Computer Networks: Motivation, Formulation, and Example, IEEE/ACM Transactions on Networking, **1**(6), (1993)
4. Yang, G., Huang D., Xie F., T.G.: a market-oriented grid computing system with fine-grained parallelism, In Proc. of Chinese National Computer Conference, (2003)
5. Thomas J. Hacker, Brian D. Athey. Account Allocations on the Grid (draft), <http://people.nas.nasa.gov/~thigpen/accounts-wg/Documents/>
6. Bill T., Tom H., Distributed Accounting on the Grid (draft), <http://people.nas.nasa.gov/~thigpen/accounts-wg/Documents/>
7. Foster I., Roy A., Sander V., A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation, In Proc. of Eight International Workshop on Quality of Service, (2000) 181-188
8. Dyer, Stephen P., The Hesiod Name Server, In Proc. of USENIX Technical Conference, Dallas, Texas, Winter, (1988)
9. J Epema D. H., Miron L, Van Dantzig R., Evers X. , and Pruyne J., A Worldwide Flock of Condors: Load Sharing among Workstation Clusters, Journal on Future Generations of Computer Systems, **12**, (1996)
10. Nirav H. Kapadia, Jose' A. B. Fortes, PUNCH: An Architecture for Web-Enabled Wide-Area Network-Computing. Cluster Computing, The Journal of Networks, Software Tools and Applications; special issue on High Performance Distributed Computing, (1999)

Reputation-Aware Contract-Supervised Grid Computing*

Xiangli Qu¹, Nong Xiao¹, Guang Xiang², and Xue-Jun Yang¹

¹ School of Computer Science, National University of Defence Technology
Changsha, China, 410073
cathysmile@eyou.com

² Institute of Computer Software and Theory, Northeastern University
Shenyang, China, 110004

Abstract. Service-related information available in current Grid is usually static, limited and self-stating. The lack of evaluation from service cooperators and record of past transactions hinders information sharing, isolates entities and separates related transactions. As a result, QoS (Quality of Service) and QoP (Quality of Protection) are hard to be guaranteed. Inspired by observations from human society, we propose a reputation-aware contract-supervised Grid computing model. The adoption of reputation targets at providing an evaluating mechanism, enhancing predictability from past experiences, promoting efficient, secure and reciprocal cooperation between entities, and forming a benign cycle in Grid computing. And contract is introduced to be a supervising mechanism. By this means, a context sensitive, deception detectable, criteria clear, and bias correctable reputation computing will be achieved. Besides, this scheme also develops some kind of CBR (Case-Based Reasoning) capabilities.

1 Motivation

The concept of Grid as an infrastructure is important because it is concerned, above all, with large scale pooling of resources, regardless of computer cycles, data, sensors, or people, undertaken in a transparent and pervasive manner [1]. From the mainstream point of service computing, each resource in Grid can be seen as a Grid service. Grid service is inherently dynamic, heterogeneous, and varied. Confronted with so many unknown services, there must be some practical mechanisms to evaluate them and some referable criteria to distinguish them. Yet, service-related information available in current Grid is very limited. Though WSDL emerges to be a solution, the information expressed is usually static and self-stating, with no evaluation from cooperators, no memory of past experiences and no records of historic transactions. In fact, this lack of dynamic information sharing results in isolated entities, disjoint transactions and unpredictable future behaviors. Confined by this limitation, many service-centric problems become hard to tackle: service match is inefficient; service security is unreliable; service quality is difficult to guarantee...

* This work is partially supported by the National 863 High Technology Plan of China under the grant No. 2002AA1Z201, China 863 OS New Technologies Research under the grant No. 2003AA1Z2060 and the Grid Project sponsored by China ministry of education under the grant No. CG2003-GA00103.

To seek a way out, we'd like to have a thought of human society. In our daily life, credit records play an important role in promoting cooperation and establishing trust relationships. While in commercial world, brand is the life of a company. The rationale in common is that reputation is an important piece of information to be exploited. Inspired by this observation, we have the idea of introducing reputation into Grid computing to bridge entities, to relate transactions, to share experiences, to leverage predictability and to maximize information sharing. And to be a workable solution, we develop a contract-supervised mechanism, which clarifies reputation context and evaluation criteria.

The rest of this paper is structured as follows: In section 2 related work is discussed and a simple introduction is presented; In section 3 we give a brief illustration of the basic model; In section 4, related components and interactions are detailed, with a case study included; finally in section 5, we conclude the paper.

2 Related Work and Introduction

Reputation is not completely new to cyberspace, especially in e-commerce, online-community and multi-agent systems. In [4], previous reputation mechanisms and related technologies are thoroughly summarized. In such domains, reputation is usually defined as the amount of trust inspired by a particular person in a specific setting or domain of interest [5].

As far as Grid is concerned, there is something different. In [3], reputation is defined as an expectation of an entity's behavior based on other entities' observations or information about the entity's past behavior within a specific context at a given time. In [2], reputation refers to the value attributing to a specific entity, including agents, services, and persons in the Grid, based on the trust it exhibited in the past. [2] proposes a reputation management framework *GridEigenTrust* to facilitate efficient resource selection in Grid, which combines two known concepts (a) using eigenvectors to compute reputation and (b) integrating global trust. Though this approach exploits some features of VO and fits for Grid environment to some extent, to be a workable and powerful Grid service, it has the following limitations:

- It only focuses on evaluating reputations of resource providers, with no consideration of requestors' reputations. As a result, half information is lost. In fact, reputation should be mutually evaluated through one transaction;
- Reputation context is vague;
- Its applicable scenario is rather limited: just confined to resource selection. In fact, reputation has great potential and broad scope in Grid;
- There is no measure to detect reputation deceptions;
- There is no criterion to for reputation evaluation. Therefore, reputation evaluated is bias-prone.

To overcome the above shortcomings and give reputation utilization a full scope in Grid computing, we put forward this reputation-aware contract-supervised Grid computing model. Our target is to make reputation evaluated mutually, utilized everywhere and attached to everything: what reputation is to Grid computing is just what credit record is to human beings. With the adoption of contract, a context sensitive, deception detectable, criteria clear and bias correctable reputation computing will be

achieved. Meanwhile, the combination of contract and reputation provides a referable case repository for Grid computing, which endows Grid with some kind of CBR (Case-Based Reasoning) capabilities.

In our opinion, reputation reflects to what extent an entity's behavior accords with its promises. Reputation cannot be self stated, but can only be rated by cooperators. Therefore, in this paper we define reputation as: an expected cooperation-satisfactory degree of an entity rated by its cooperators according to its performance in carrying out specific service contracts. Since we emphasize on mutual reputation evaluation and each service inevitably relates to a provider and a requestor, we'd like to classify reputations into 2 kinds:

- Service reputation: this kind of reputation relates to a service provider, evaluated from its past performance when providing service under specific contracts, reflects to what degree the provider can be expected to fulfill its declarations in a service contract.
- User reputation: this kind of reputation relates to a service requestor, evaluated from its past performance when using certain service under specific contract, reflects to what degree the requestor can be expected to fulfill its declarations in a service contract.

3 Basic Model

In order to put our ideas into realization, we introduce two Grid services: Grid Reputation Service (GRS) and Grid Contract Service (GCS) into Grid computing. The basic model is depicted in Fig. 1:

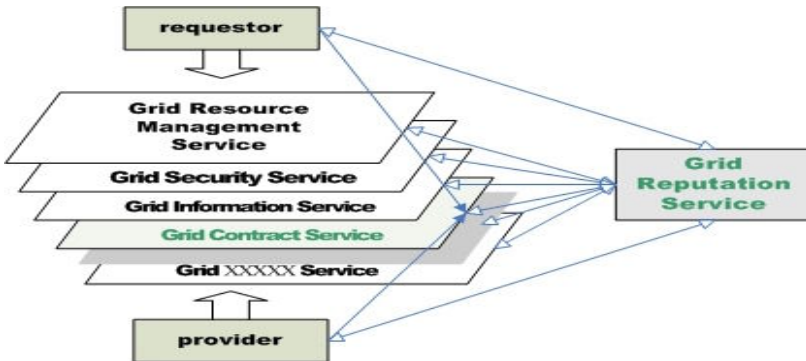


Fig. 1. Basic Model

As can be seen from Fig. 1, each entity within Grid, no matter basic services such as Grid Resource Management Service, Grid Security Service or customized services such as Grid Contract Service, has a bidirectional association with GRS. On one hand, they will benefit from GRS: In a service transaction, each entity acts either as a requestor or as a provider. As a requestor, the entity will need GRS to select a reliable and suitable provider. While as a provider, the entity will need GRS to evaluate its counterpart's trustworthiness, and accordingly enforce specific security requirements

and set up appropriate executing environments. On the other hand, they will subject to GRS's reputation evaluation. In this way, reputation indeed goes everywhere and with everything, forming a reputation-aware Grid computing environment.

Yet, reputation is more or less bias-prone, different entity might have different evaluation towards the same service quality. To minimize this deviation and enable a relatively fair evaluation, Grid Contract Service is proposed to give a hand. To start a specific service transaction, the two participants will first negotiate a specific contract through GCS. After transaction, reputation will be mutually evaluated based on corresponding contract fulfillments. Verified by GCS, each participant's reputation: service reputation and user reputation will be reported to GRS. Then, GRS will carry out an analysis and aggregation of the reported reputation, and finally deposit it and disseminate it on demand.

4 Components and Interactions

4.1 Contract Related

In our model, each service transaction is associated with a contract C_k . Contract is a key component signed by GCS, which provides a relatively objective criterion to evaluate a participant's reputation in a specific service transaction. By means of contract, a detailed list of requirements for both participants is recorded and agreed by both. Thus each contract has two parts: one is for the provider, and the other is for the requestor. On the provider's part, requirements will include: network bandwidth requirements, response time requirements, security requirements and so on. On the requestor's part, requirements will include: maximum disk space occupation, cleanup when logging out, security requirements and so on. Since each specific service transaction might lay different emphases on different requirements, for each requirement R_i in the contract the participants will negotiate a corresponding weighted coefficient W_i , where $W_i \in (0,1)$ and $\sum_{i=1}^n W_i = 1$ (n is the number of total requirements for a participant). Meanwhile, to enable a mutually recognized reputation computing method, a specific reputation computing function $\phi_i(R_i, P_i)$ is also negotiated for each specific requirement R_i , herein P_i is the real performance of the participant and $\phi_i(R_i, P_i) \in [0,1]$. After each transaction, a mutual reputation computing will be carried out. Participant I_p will compute its counterpart I_q 's reputation $Rep(I_p, I_q, C_k)$ according to Equation (1):

$$Rep(I_p, I_q, C_k) = \sum_{i=1}^n W_i \phi_i(R_i, P_i), \quad (n \text{ is the number of total requirements for } I_q \text{ in contract } C_k) \quad (1)$$

In our model, on finishing a service transaction, both participants should submit a contract fulfillment report including real performance of each specified requirements and the reputation computing result to the GCS where they signed the contract. In

order to supervise this activity, a contract-reporting deadline t is also negotiated in the contract. To summarize, a contract will include the following contents:

- 1) Requirements for requestor and provider R_i ;
- 2) Specific reputation computing function for each requirement $\phi_i(R_i, P_i)$;
- 3) Weighted coefficient for each requirement-related reputation evaluation w_i ;
- 4) A specific report-submitting deadline t

With the above information, it is no exaggeration to say that reputation computing under such contract is context sensitive and criteria clear.

4.2 GCS Related

GCS is provided to ease contract negotiation and supervise contract fulfillment. It is composed of four components: Contract Negotiation, Contract Validation, Policy Repository and Contract Repository, as is depicted in Fig. 2:

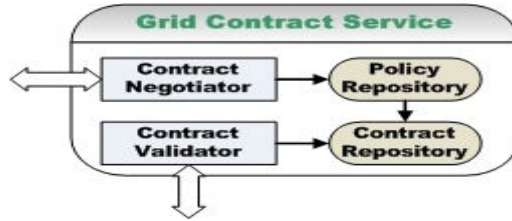


Fig. 2. GCS Components

Contract Negotiation component together with Policy Repository is supplied to help cooperators negotiate a proper contract. Policy Repository is mainly filled with various kinds of reputation computing functions $\phi(R, P)$ tailored for specific requirement types, such as time urgent, data intensive and so on. This is a dynamic repository. With the growth of Contract Repository, new functions will be added to enrich the policy deposit.

Contract Validation component together with other Grid services such as Grid Job Management Service, Grid Security Service and so on is responsible for validating and tracking contract fulfillment. This is also an enhancement to detect reputation deception. On receiving a contract fulfillment report, this component will first check its validity. According to the checking result, verified reports will be deposited in the Contract Repository; forged reports will be marked and related participants' reputation will be reevaluated. If a participant misses its reporting deadline t , the Contract Validation component will look into this event and reevaluate the two participants' reputation. Finally, GCS will submit a reputation report to GRS. According to [6], to avoid reusing the same reputation result and minimizing the effect of evidence correlation, the submitted reputation report contains not only a reputation grade but also its related service contract.

4.3 GRS Related

4.3.1 Components

The proposed Grid Reputation Service consists of six components: Reputation Acquisition, Reputation Analysis, Reputation Aggregation, Reputation Update, Reputation Dissemination and Reputation Repository, as is depicted in Fig. 3:

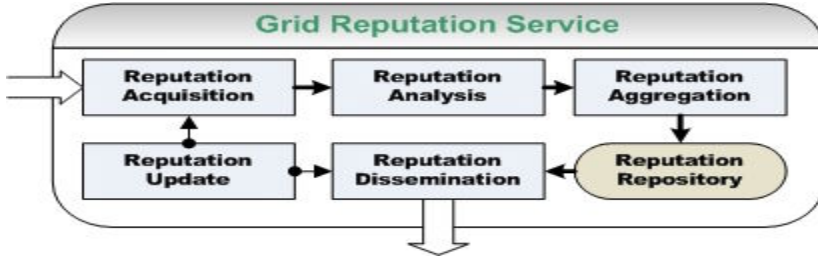


Fig. 3. GRS Components

The functionality of each component is:

- **Reputation Acquisition:** This component is responsible for acquiring evidence for reputation aggregation. Reputation evidence can be acquired flexibly through “push” and “pull” modes. The reputation report submitted by GCS mentioned above is a kind of “push” acquisition. Such reputation evidence can also be pushed directly by individual service participant. Meanwhile, this component can also actively pull reputation evidences from other GRS implementations and GCS implementations.
- **Reputation Analysis:** Since reputation evidences collected might be correlated, outdated or even forged, it is necessary to perform a thorough analysis before aggregation. Reputation Analysis component is just in charge of this job. For example, it will inquire the GCS who signed the contract of the validity of the reputation evidence. And to avoid cooperative deception, reputation evidences from the same origin within a specific period of time will be aggregated first, then used as one evidence. Thus, this is another gate to prevent reputation deception.
- **Reputation Aggregation:** As its name implies, this component is responsible for reputation classification and aggregation. We will not elaborate on the specific algorithms here for space limitation. In fact, as there exists no optimal algorithm universally applicable, we’d better provide an algorithm repository to adapt to different scenarios.
- **Reputation Dissemination:** To facilitate usage and keep reputation information latest, an entity can order reputation updates from GRS. This job is done by Reputation Dissemination component periodically or on demand.
- **Reputation Repository:** This repository is deposited with classified and verified reputation evidences, including related contracts and final grades. In fact, this is a precious record and evaluation of past transactions, which accumulates dynamic service-related information. Many Grid services such as Grid Information Service, Grid Resource Management Service, Grid Security Service and so on will benefit from it. With this repository, a service requestor will rapidly find an appropriate

and reliable provider to meet its requirements, while a provider will predict its counterpart's trustworthiness from concrete evidence, which endows Grid with a kind of CBR capability.

- **Reputation Update:** Since a specific reputation decays with time, it is necessary to perform an update after a given period of time. This update will trigger new reputation acquisition and dissemination. Afterwards, new reputation analysis and aggregation will be triggered successively.

4.3.2 Interactions

Around reputation, GRS is tightly associated with entities in Grid. Any entity in Grid can order reputation updates, query specific reputation evidence, call specific reputation computing from GRS and submit reputation reports to GRS, while GRS can pull reputation evidences from individual entities, and disseminate reputation updates to them. To be a powerful service, GRS will also need other Grid services such as Grid Security Service, Grid Resource Management Service etc. to leverage efficiency, improve accuracy and enhance reliability

Reputation computing in our model is very flexible. It can be done either locally by aggregating evidences with tailored policy, or remotely by completely relying on GRS to get a final result.

4.4 A Case Study

In this section we'll give an example illustrating how a service transaction is processed in our model. Suppose a user A wants to run a biological emulation elsewhere and is in search of such a service provider. The whole process goes as follows:

- 1) From a locally or remotely computed reputation result, A chooses a reliable service Registry R .
- 2) A sends a request to R stating its query intention. After checking A 's reputation, R decides whether to provide this service or not.
- 3) Suppose R accepts the request, they negotiate a query contract C_i with the help of GCS C .
- 4) According to A 's reputation level and specifications in C_i , R sets up corresponding executing environments, performs A 's query and responds with a list of candidate emulation providers.
- 5) According to each other's real performance and corresponding declarations in contract C_i , R and A mutually compute each other's reputation and reporte to C .
- 6) After verified the reputation report, C submits a signed report to GRS S .
- 7) On receiving the report, S first makes an analysis, then aggregates it with other evidences and finally stores it in its Reputation Repository.
- 8) For the candidate emulation providers, A computes their reputations locally or remotely. According to the reputation result, A chooses a provider E . And the next steps will repeat the similar activities as what happens from step 2) to step 7).

Noted here, from step 4), there is no fixed order between step 5) and step 8). They may occur concurrently or successively depending on specific scenarios.

5 Conclusions

In this paper, we propose a reputation-aware contract-supervised Grid computing model. This model exploits the sharing and cooperative nature of Grid computing. With no memory and no evaluation of finished service transactions in previous Grid computing, referable experiences are lost and related transactions are disjoint. Therefore, we introduce reputation as an evaluation, a record and a prediction. Both providers and requestors can benefit from this dynamic information, which enables an efficient, reliable and suitable service match, promotes cooperation between entities, enhances Grid security and enriches Grid shared information. To be a workable solution, contract is adopted to be a supervising mechanism and an evaluating criterion. As stated before, GRS and GCS cooperatively achieve a context sensitive, deception detectable, criteria clear and bias correctable reputation computing, which is the cornerstone for the whole scheme to succeed. Furthermore, contract plus reputation forms a dynamic and evolving case base for Grid, therefore a kind of CBR capability is enabled.

References

1. Omer F. Rana and David W. Walker: Service Design Patterns for Computational Grids. Patterns and Skeletons for Parallel and Distributed Computing, 2003
2. Beulah Kurian Alunkal, Ivana Veljkovic, Gregor von Laszewski, and Kaizar Amin: Reputation-Based Grid Resource Selection. Workshop on Adaptive Grid Middleware (AGridM 2003), September 28, 2003.
3. Farag Azzedin and Muthucumaru Maheswaran: Evolving and Managing Trust in Grid Computing Systems. Proceedings of the 2002 IEEE Canadian Conference on Electrical Computer Engineering.
4. G. Zacharia and P. Maes: Trust Management through Reputation Mechanisms. Applied Artificial Intelligence, 14: 881-907, 2000.
5. Marsh, S. P.: Formalising trust as a computational concept. Ph.D. Thesis, 1994.
6. Michael Schillo, Petra Funk and Michael Rovatsos: Using Trust for Detecting Deceitful Agents in Artificial Societies. Applied Artificial Intelligence, 14:825--848, 2000.

The Analysis of Efficient Monitoring Grid Traffic with Flow Conservation Equation

Xianghui Liu¹, Jianping Yin¹, Xicheng Lu¹, Zhiping Cai¹, and Jianmin Zhao²

¹ School of Computer Science, National University of Defense Technology
Changsha City, Hunan Province, 410073, PRC
LiuXH@tom.com

² School of Computer Science, Zhejiang Normal University
Jinhua City, Zhejiang Province, 321004, PRC

Abstract. Monitoring grid traffic is a key step for grid resource management, the problem of efficient monitoring for the grid traffic is to reduce the generation overhead communication as more as possible regarding as the problem by identifying the minimum weak vertex cover set for a given graph $G(V,E)$ which represents the topology of a grid. An approximation algorithm to find out the weak vertex set is presented and it is proved that the algorithm has a ratio bound of $2(\ln d + 1)$, where d is the maximum degree of the vertices in graph G . Then it is showed that the running time of the algorithm is $O(|V|^2)$.

1 Introduction

Monitoring network characteristics such as grid traffic is critical to the performance of grid application[1,2]. Distributed applications rely on timely and accurate information about the available bandwidth to make informed decisions about where to send data for computation and how to adapt to changing conditions. In the current grid computing environments, with the absence of any type of bandwidth reservation mechanisms, a tool for monitoring traffic in grid computing environments is needed that is part of the grid infrastructure itself [1].

Grid traffic is measured either by actively injecting data probes into the network or passively monitoring existing traffic. The active approach may cause competition between application traffic and the measurement traffic, reducing the performance of useful applications. Because the passive approach avoids the problem of contention by passively observing and collecting network characteristics, we have in-depth research in this approach [3,4]. Currently most passive monitoring approaches typically assume that the monitoring instrumentation can be either intelligently distributed at different points or placed at the endpoints of the end-to-end path whose characteristics are of interest [5,6]. As modern grid traffic monitoring process requires more data to be collected and at much higher frequencies [1]. Then the overhead that monitoring method imposes on the underlying router can be significant and adversely impact the router's throughput and severely impact its performance [7,8].

In this paper, we focus efficient monitoring for the grid traffic on reducing the generation overhead communication as more as possible. And the problem of efficient monitoring is regarded as the problem to find out the minimum weak vertex cover set

for a given graph. Once the topology of a grid is acquired, the approximation algorithm proposed in this paper can generate rapidly and automatically a set of monitoring nodes for the network. There are fewer members in the set, so the overhead for monitoring the grid traffic is not high.

2 Problem Descriptions

Definition 1: Given an undirected graph $G(V,E)$ which represents the topology of a grid, where V represents the set of nodes, E represents the set of edges between two nodes, we say $S \subseteq V$ is a traffic monitoring set of G , if monitoring the traffic of those edges that are incident on nodes in S is sufficient to infer the traffic of every edge in E .

The goal of efficiently monitoring traffic of graph G is to find out the minimum traffic monitoring set of G . Though we can determine a traffic monitoring set by using the minimum vertex cover set, the minimum vertex cover problem is NP-hard and up to date there is no polynomial algorithm. Moreover, the Traffic Monitoring Set got by the minimum vertex cover algorithm may not necessarily be the best, because if the nodes represent the routers in a network, we still have two constraints to be used:

- (1) $\forall v(v \in V \rightarrow Degree(v) \geq 2)$, where $Degree(v)$ denotes the degree of node v .
- (2) $\forall v(v \in V \rightarrow \sum_{u \in V} f(v,u) = 0)$, where $f(v,u)$ is the traffic from node v to node u ,

which can be positive or negative.

Constraint (2) is flow conservation equation. In reality the flow conservation equation only holds approximately, since there can be (a) extra traffic directed to/from the router, (b) multicast traffic that is replicated along many output interfaces, and (c) delayed and dropped packets in the router. Several measurements over backbone routers have showed that traffic conservation holds with a relative error that is consistently below 0.05% [3,9].

3 Weak Vertex Cover

Definition 2: Given an undirected graph $G(V,E)$ representing the topology of a grid,, where $\forall v(v \in V \rightarrow Degree(v) \geq 2)$ holds, we say the subset S of V is a weak vertex cover set of G [7], if and only if every edge in G can be marked by performing the following three steps:

- (1) Mark all edges that are incident on vertices in S .
- (2) Mark the edge if it is the only unmarked edge among all the edges that are incident on the same vertex.
- (3) Repeat step (2) until no new edge can be marked.

The weak vertex cover set of G is a traffic monitoring set of G under the constraint of traffic conservation equation. The traffic of all edges marked in step (1) can be monitored by measurement and the traffic of the edge marked in step (2) can be calculated by using flow conservation equation.

Lemma 1: Given an undirected graph $G(V, E)$, where $\forall v(v \in V \rightarrow Degree(v) \geq 2)$ holds, then $S \subseteq V$ is a Weak vertex cover set of G , if and only if $G' = (V', E')$ is a forest, where $V' = V - S$ and $E' = \{(u, v) \mid (u, v) \in E \wedge u \in V' \wedge v \in V'\}$.

Proof: (Necessity) If $G' = (V', E')$ is not a forest, G' must contain a cycle. Let $v_1, e_1, v_2, e_2, \dots, v_m, e_m, v_1$ be a cycle in G' . For $v_i (1 \leq i \leq m)$ is not in S , $e_i (1 \leq i \leq m)$ can't be marked in step (1) of definition 2. Unmarked e_i is incident on v_i and v_{i+1} (v_1 when $i = m$), moreover, e_{i-1} (e_m when $i = 1$) and e_{i+1} (e_1 when $i = m$) incident on v_i and v_{i+1} (v_1 when $i = m$) respectively are unmarked too, so $e_i (1 \leq i \leq m)$ can't be marked in step (2) of definition 2. This contradicts the assumption that S is a weak vertex cover set of G .

(Sufficiency) In step (1) of definition 2, we can mark all the edges in $E - E'$. If $G' = (V', E')$ is a forest, in step (2) of definition 2, we can repeatedly mark the edge incident on a leaf of a tree in the forest G' until all the edges in E' have been marked. So S is a weak vertex cover set of G . ■

Theorem 1: Find the minimum weak vertex cover set of an undirected graph $G(V, E)$, where $\forall v(v \in V \rightarrow Degree(v) \geq 2)$ holds, is NP-Complete.

Proof: We reduce the vertex cover problem VC to this weak vertex cover problem.

Instance: An undirected graph D , and an integer k . And Question: Is there a set S of at most k vertices in D such that S is weak vertex cover set of graph D ?

Instance: A graph G and an integer k . And Question: Is there a vertex cover in G with at most k vertices?

Firstly, this weak vertex cover problem is an NP problem since a YES instance can be checked by a given set with at most k vertices in polynomial time.

To prove this problem is NP-complete, we reduce VC to this problem: Let graph G be an instance of VC. Construct an undirected graph D , as an instance of weak vertex cover set as follows: Replace every edge uv in G by a pair of edges uv and vu . This transformation can obviously be done in polynomial time. Now we prove that (G, k) is a yes instance of VC if and only if (D, k) is a yes instance of weak vertex cover.

If (G, K) is a YES instance of VC, then G has a vertex cover S with at most k vertices. By the construction of D , every arc in D has at least one end in S . Let $v_1 v_2 \dots v_k v_1$ be a cycle in D . Since $v_1 v_2$ is an edge, either v_1 or v_2 is in S . Hence, every cycle has at least one vertex in S . Since S has at most k vertices, D has a weak vertex cover set with at most k vertices. In other words, D is a YES instance of weak vertex cover.

On the other hand, assume (D, k) is a YES instance of weak vertex cover. Then D has a set S with at most k vertices such that every cycle in D has at least one vertex in S . In particular, for every edge pair uv and vu , as a cycle of length 2, either u or v is in S . Since such a pair corresponds to an edge in G , every edge uv in G has an end in S . Therefore, S is a vertex cover of G . Since S has at most k vertices, G has a vertex cover with at most k vertices. In other words, G is a YES instance of VC. ■

Below a greedy algorithm is presented. After inputted an undirected graph $G(V, E)$, the algorithm can output a weak vertex cover set U of G .

Algorithm CaculateWeakVertexCoverSet (Graph G)

```

{
1      U =  $\Phi$  ;
2      i = 1;
3       $G_i = G$ ;
4      while (the vertex set of  $G_i$  is not empty)
      {
5          select a vertex  $v_i$  in  $G_i = (V_i, E_i)$  such
that Degree( $v_i$ ) is maximum;
6           $U = U + \{v_i\}$ ;
7           $V' = V_i - \{v_i\}$ ;
8           $E' = E_i - \text{Adj}(v_i)$ ;
9          i = i + 1;
10         repeat remove all the vertices with de-
gree 0 or 1 and all the edges that are incident on them
from  $G' = (V', E')$  until no new vertex or edge can be
removed. And let  $G_i$  be the resulting graph;
      }
11      return Set U;
}

```

4 Algorithm Analysis

Let U^* be the minimum weak vertex cover set of graph $G(V, E)$, $\overline{U^*} = V - U^*$, $G_i(V_i, E_i)$ be the sub-graph handled in the i th loop of the algorithm. U represents the approximation solution of U^* produced by the algorithm. $|U|=m$, $U=\{v_1, v_2, \dots, v_m\}$. And $d_i(v)$ be the degree of vertex v in sub-graph G_i , $d_X(V)$ be the number of edges of which one vertex is v and another is in set X . Because greedy strategy is used in the algorithm, $d_i(v_i) \geq d_i(v_j)$ holds for any vertex v_i and $v_j (1 \leq i < j < m)$ in U . Let $U_i^* = U^* \cap V_i$, $\overline{U_i^*} = \overline{U^*} \cap V_i$, $V_{m+1} = \Phi$, $S_i = V_i - V_{i+1}$, $(1 \leq i \leq m)$. We have the following lemma 2.

Lemma 2: $\sum_{j=i}^m d_j(v_j) \leq 2 \sum_{v \in U_i^*} d_i(v)$ for all $(1 \leq i \leq m)$. (Proof Ommitted)

Theorem 2: Let U^* be the minimum weak vertex cover set of graph $G(V, E)$, U be the approximation solution of U^* produced by the algorithm. Then $|U| \leq 2H(d)U^*$,

where $H(x) = \sum_{i=1}^x 1/i$, $d = \max_{v \in V} \{Degree(v)\}$.

Proof: Assume that the cost to put a vertex from graph $G_i(V_i, E_i)$ into weak vertex cover set is 1, and the cost is uniformly distributed to all the edges that are incident on the vertex. Then the cost of every edge incident on the vertex v_i is $c_i = 1/d_i(v_i)$ for

$$1 \leq i \leq m. \text{ So } |U| = \sum_{i=1}^m c_i d_i(v_i) = c_1 \sum_{i=1}^m d_i(v_i) + \sum_{i=2}^m (c_i - c_{i-1}) \sum_{j=i}^m d_j(v_j).$$

For $2 \leq i \leq m$, by the greedy strategy of the algorithm to select v_{i-1} and the monotony of d_i about i , we have $d_i(v_i) \leq d_{i-1}(v_i) \leq d_{i-1}(v_{i-1})$. So $c_i \geq c_{i-1}$. By the definition of c_i , we have $c_i \geq 0$. By the construction of V_i in the algorithm, V_i can be decomposed

into the union of several disjoint sets: $V_i = \sum_{j=i}^{m-1} (V_j \setminus V_{j+1}) \cup V_m$. And

$$U_i^* = \sum_{j=i}^{m-1} (U_j^* \setminus U_{j+1}^*) \cup U_m^*. \text{ Using lemma 2, we have:}$$

$$|U| \leq 2 \sum_{j=1}^{m-1} \sum_{v \in U_j^* \setminus U_{j+1}^*} \left(\sum_{i=1}^{j-1} (d_i(v) - d_{i+1}(v)) c_i + d_j(v) c_j \right) + 2 \sum_{v \in U_m^*} \left(\sum_{i=1}^{m-1} (d_i(v) - d_{i+1}(v)) c_i + d_m(v) c_m \right).$$

Notice $U^* = U^* \cap V = U^* \cap V_1 = U_1^* = \sum_{j=1}^{m-1} (U_j^* \setminus U_{j+1}^*) \cup U_m^*$. We have:

$$|U| \leq 2 \sum_{v \in U^*} \sum_{i=1}^{s(v)} (d_i(v) - d_{i+1}(v)) c_i = 2 \sum_{v \in U^*} \sum_{i=1}^{s(v)} (d_i(v) - d_{i+1}(v)) / d_i(v).$$

For $1 \leq i \leq s(v)$, by the greedy strategy of the algorithm to select v_i , we have $d_i(v) \leq d_i(v_i)$. By the monotony of d_i about i , we have $d_{i+1}(v) \leq d_i(v)$. That is to say $d_i(v) - d_{i+1}(v) \geq 0$. So $2 \sum_{v \in U^*} \sum_{i=1}^{s(v)} (d_i(v) - d_{i+1}(v)) / d_i(v) \leq 2 \sum_{v \in U^*} \sum_{i=1}^{s(v)} (d_i(v) - d_{i+1}(v)) / d_i(v)$. And for

integer a and b , if $a \leq b$, then $H(b) - H(a) = \sum_{i=a+1}^b 1/i \geq (b-a)/b$.

$$\begin{aligned} \text{So } |U| &\leq 2 \sum_{v \in U^*} \sum_{i=1}^{s(v)} (H(d_i(v)) - H(d_{i+1}(v))) = 2 \sum_{v \in U^*} (H(d_1(v)) - H(d_{s(v)+1}(v))) \\ &= 2 \sum_{v \in U^*} (H(d_1(v)) - H(0)) = 2 \sum_{v \in U^*} (H(d_1(v))) \leq 2H(d)U^* \end{aligned}$$

By $H(d) \leq \int_1^d (1/x) dx + 1 = \ln d + 1$, we have: $|U| \leq 2H(d)U^* \leq 2(\ln d + 1)U^*$. ■

Theorem 3: The running time of the algorithm on any graph $G(V, E)$ is $O(|V|^2)$.

Proof: Assume that the set U is represented by a linked list L and the graph G by an adjacency matrix A . Array D is used to record the degree of every vertex in graph and linked list Q to record the numbers of vertices with degree 1 in graph. The execution time of statements ① and ② are constant; The running time of statement ③ is

$O(|V|^2)$; Initialization of array D takes time $O(|V|^2)$ and initialization of linked list Q takes constant time. Because at least one vertex is removed in every loop, the times of the loop is at most $O(|V|)$. With the aid of array D , the time to run statement ⑤ once is $O(|V|)$ and the time to run statement ⑥ once is constant. The execution of statement ⑦ and ⑧ is as follows. Set $D[i]=0$. Then scan the i th row of adjacency matrix A . For any $j(1 \leq j \leq |V|)$, if $A[i,j]=1$, then set $A[i,j]=A[j,i]=0$ and $D[i]=D[i]-1$. If $D[i]=1$, put j into the linked list Q . So the time to run them once is $O(|V|)$. The time to run statement ⑨ once is constant. The execution of statement ⑩ is as follows. When the linked list Q is not empty, j is taken out from Q repeatedly. Set $D[i]=0$. Then scan the j th row of adjacency matrix A . For any $k(1 \leq k \leq |V|)$, if $A[i,k]=1$, then set $A[i,k]=A[k,j]=0$ and $D[k]=D[k]-1$. If $D[k]=1$, put k into the linked list Q . For every j taken out from Q , the time to complete the above work is $O(|V|)$. On the other hand, during the whole execution of the algorithm, the number of every vertex is put into the linked list Q at most once, so the total of numbers taken out from Q is at most $|V|$. Hence, the whole time to run statement ⑩ is $O(|V|^2)$. The termination condition of the loop can be implemented as follows. After v_i is selected in statement ⑤, if $D[i]=0$, then terminate the loop, otherwise continue. To sum up, the running time of the algorithm on any graph $G(V,E)$ is $O(|V|^2)$. ■

5 Conclusion

In our experiments, we used two different network generators to generate random networks with different characteristics. One generator was based on waxman model, the other on power-law model[10]. Experiments show we can exploit extra useful information and reducing the monitoring nodes.

Noticing that the constraint of flow conservation equation only holds approximately, our research in the future will focus on estimating the influence of the approximation error on the flow of edges calculated by this method.

References

1. Bruce B. Lowekamp, Nancy Miller, Roger Karrer, Thomas Gross, and Peter Steenkiste. Design, Implementation, and Evaluation of the Remos Network Monitoring System. *Journal of Grid Computing* 1(1) 2003: 75-93
2. G. He and J. C. Hou. On Exploiting Long Range Dependence of Network Traffic in Measuring Cross Traffic on an End-to-end Basis. *IEEE Infocom*, No. 1, 2003:1858-1868
3. A.B.Downey, Using Path Char to Estimate Internet Link Characteristics. In *Proceeding of the ACM SIGCOMM'99 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*. Cambridge, 1999:241-250
4. K.Lai and M.Baker, Measuring Bandwidth. in *Proc. of IEEE INFOCOM'99*. March 1999
5. S.Jamin, C.Jin, Y.Jin, Y.Raz, Y.Shavitt and L.Zhang. On the Placement of Internet Instrumentation. In *Proceeding of IEEE INFOCOM'2000*. Tel-Aviv, Israel, 19(1), March 2000 :295-304
6. R. CBceres, NG Duffield, A. Feldmann, J. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. Kalmanek, B. Krishnamurthy, D. Lavelle, PP Mishra, KK Ramakrishnan, J. Rexford, F. True, and JE van der Merwe. Measurement and analysis of IP network usage and behavior. *IEEE Communications Magazine*. 38(5),2000:144-151.

7. Y.Breibart, C.Y.Chan, M.Carofalakis, R.Rastogi, A.Silberszhatz Y.Breibart, C.Y.Chan, M.Carofalakis, R.Rastogi, A.Silberszhatz. Efficiently monitoring bandwidth and latency in IP network. In Proceeding of IEEE INFOCOM 2001. Anchorage, Alaska, USA ,20(1), April 2001 :933-942.
8. XH Liu, JP Yin, XC Lu, JM Zhao. A monitoring model for link bandwidth usage of network based on weak vertex cover. Journal of Software(Chinese), 15(4),2004:545-549.
9. Habib A, Khan M, Bhargava B. Edge-to-Edge measurement-based distributed network monitoring. Technical Report, CSD-TR-02-019, Purdue University, 2002.
10. Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The origin of power-laws in Internet topologies revisited. In IEEE INFOCOM, June 2002.

Dynamic Semantic Clustering Approach for Web User Interest

Jiu Jun Chen, Ji Gao, Bei Shui Liao, and Jun Hu

College of Computer Science, Zhejiang University, Hangzhou 310027, Zhejiang, China
rackycjj@163.com

Abstract. To extract a dynamic interest model, we proposed an approach to mine multilayer interests from navigational behavior and favorite pages of web user. Our works based on the ideas that changes of the user interests can be tracked from his or her navigational behavior, and the changeable interests might derivate from the same kind of interests at a higher abstraction level. Markov user model (MUM) is used to learn the navigational characters of web user. Based on both MUM and user's favorite pages, dynamic semantic mining approach is designed to construct multilayer user interest, which represents the user's specific as well as general interests. The higher-level interests are more general, and the lower-level ones are more specific. The model implements in our example website to mine the dynamic continuum of long-term to short-term interests of web user. It proves that the results are good.

1 Introduction

User interest can be used to create a more robust web service. It can help in filtering the information, improving the content and design of the website, and customizing the service to the needs of specific users [1, 2, 3]. Web user interest is changeable, that is, one user can exhibit different kinds of interests at different times; different user's interests are also different [4, 5]. In order to track the dynamic interest of web user, a new mining method and continuum interest representation are necessary.

Our work in this paper is to study a new semantic clustering approach to learn a dynamic multilayer user interests from favorite web pages. We believe that the variety of user interest can be reflected from his or her navigational behavior, and the different interests might be motivated by the same kind of interests at a higher abstraction level. The approach is built in three steps: (a) finding the favorite pages visited by the user based on Markov user model; (b) extracting words from pages and grouping features into cluster; (c) mining dynamic topic about interests according to the clusters and representing as a dynamic multilayer user interest (MUI).

The rest of the paper is organized as follows: the next section presents a survey of related work in clustering algorithm and user profiling methods. In section 3, we first introduce the Markov user model. The dynamic semantic clustering algorithm is then designed. Section 4 discusses experimental results and the conclusions are presented in the final section.

2 Related Works

Clustering method can be viewed as unsupervised learning from a given dataset. It can be classified into two categories: hierarchical clustering and center-based clustering. Hierarchical clustering finds the clusters by initially assigning each object to its own cluster and then repeatedly merging similar clusters together until a certain stopping condition is met [6]. It is resulted in the form of tree. The main advantage of hierarchical clustering lies in its ability to provide a view of data at multiple levels of abstraction. But we should determine where to cut the dendrogram to produce clusters. In center-based algorithms, all objects start in one cluster initially. It is repeatedly partitioned into either a pre-determined (e.g., k-means clustering) or an automatically derived number of clusters (e.g., X-means clustering) [7, 8, 9]. It uses a global criterion function whose optimization produces the entire clustering process, but it is susceptible to a local optimum. Our semantic clustering is enhanced and robust, whose input is dynamic and incremental. It can remove weak relations and only needs to cluster strongly connected features.

Some works about user profiling have been studied. Probabilistic clustering of individuals with mixtures of Markov chains is used to learn the web user pattern [10, 11, 12, 13]. SCML, a concept learning algorithm that extracts some concepts in a set of data, is introduced by Perkowitz and Etzioni [1]. Kominek and Kazman [14] designed a multimedia information retrieval system, named Jabber, to realize access multimedia through concept clustering technology. Barbu and Simina [15] presented an algorithm for learning incrementally the profiles of a user, based on an initial user profile and on user's queries using probabilistic latent semantic analysis. In [16], a news agent called News Dude uses a multi-strategy machine learning approach to create separate models of a user's short-term and long-term interests. Unlike News Dude, we model a dynamic continuum of long-term to short-term interests, which can track the changeable of user interest and update the interest topics automatically.

3 Dynamic Semantic Clustering

3.1 Mathematical Modeling

Because web user is largely unknown from the start, and may change during the exploration, Markov user model is constructed to learn the favorite pages of web user according to his or her navigational activities. Some items in the model are defined as follows.

Definition 1. State A state is defined as a collection of one or more pages of the website with similar functions. Besides n functional states, the model contains other two especial states, Entry state and Exit state;

Definition 2. Transition probability p_{ij} A transition occurs with the request for one page that belongs to state j while the user resides in one of the pages belonging to state i . Transition probability is the probability of transition from state i to state j ;

Definition 3. Mean Staying time $\bar{t}_{ij}$ It is the mean time which the process remains in one state before making a transition to another state;

Definition 4. Favorite f_{ij} It is defined as the evaluation of the interest level of the state, and it integrates the influence of transition probability and mean staying time. We supposed that, (a) if there are n kinds of different transition to leave one state, the state that has higher transition probability reveal user interest; (b) if there are n kinds of different transition to leave one state, those states that have long staying time reveal user interest.

But one problem should be studied that some pages that is only to be utilized the links of a page to another page may also have many visited times. In our model, we used Favorite definition to prevent from only mining visited states with high probability and low staying time. It defined as formula (1)

$$\left\{ \begin{array}{ll} f_{ij} = \frac{p_{ij} \times \bar{t}_{ij}}{(\sum_{j=2}^{n+1} p_{ij})(\sum_{j=2}^{n+1} \bar{t}_{ij})/(n)^2} & i, j \in (2, n+1) \\ f_{1j} = 1 & j \in (2, n+1) \\ f_{l(n+2)} = 0 \\ f_{il} = 0 & i \in (1, n+2) \\ f_{(n+2)j} = 0 & j \in (2, n+1) \\ f_{i(n+2)} = 1 & i \in (2, n+2) \end{array} \right. \quad (1)$$

As shown in formula (2), a set of four elements defines a discrete Markov user model. It is a state to state matrix, which used to describe the favorite level of each state or page based on web user's navigational behavior dynamically. p_{ij} and $\bar{t}_{ij}$ in the user model are described in algorithm1.

$$UserModel = \{state_{ij}, f_{ij}, p_{ij}, \bar{t}_{ij} : i, j \in (1, n+2) \} \quad (2)$$

Algorithm 1. Generating Transition Probabilities and Mean Staying Time

Step 1. For the first request for state s in the session, add a transition from Entry state to the state s , and increment $TranstionCount_{0s}$ in a matrix $TranstionCount[i, j]$ by 1, where $TranstionCount[i, j]$ is a matrix to store the transition counts from state i to state j ;

Step 2. For the rest of user' requests in the session, increment the corresponding transtion count of $TranstionCount_{i,j}$ in the matrix, where i is the previous state and j is the current state;

Step 3. For the last page request in the session, if the state is not the explicit exit state then add a transition from the state to exit state and increment $TranstionCount_{s,(n+2)}$ value by 1;

Step 4. Divide the row elements in matrix $TransitionCount[i, j]$ by the row total to generate transition probability matrix P , whose element is $p_{i,j}$:

$$p_{i,j} = \frac{TransitionCount_{i,j}}{\sum_j TransitionCount_{i,j}} \quad (3)$$

Step 5. To find out the time spent in state i before the transition is made to state j for any transition from state i to state j , except the transition from entry state, and the transition to the exit state. If this time belongs to the interval k then, increment $StayTimeCount_{i,j,k}$ by 1 in a three-dimensional matrix $StayTimeCount[i, j, m]$, where, $StayTimeCount_{i,j,k}$ is the number of times the staying time is in the interval k at state i before the transition is made to state j .

Step 6. Find out the interval total for each transition from state i to state j in $StayTimeCount[i, j, m]$. Divide frequency count in each interval with the interval total to find out the probability of occurrence of the corresponding intervals. Repeat this to generate $StayTime Probability[i, j, m]$. whose elements defined as follows:

$$StayTime Probability_{i,j,m} = \frac{StayTimeCount_{i,j,m}}{\sum_m StayTimeCount_{i,j,m}} \quad (4)$$

Step 7. Multiply each interval with the corresponding probability to generate mean staying times ($\bar{t}_{ij}$), which is the elements of matrix $\bar{T}$.

$$\bar{t}_{ij} = \sum_m m \times StayTime Probability_{i,j,m} \quad (5)$$

3.2 Semantic Clustering

Semantic clustering algorithm is used to construct multilayer user interest. The four phases are studied next.

Step 1: Preprocessing

In this phase, some words in states or pages visited by web user are extracted. Actually we extract only nouns from the favorite pages and simplify the problem. By stemming techniques, different forms of the same words are converted to their root.

Step 2: Calculating the Word Similarity Matrix

Similarity matrix is used to measure closeness between a pair of words. We assume that words occurring close to each other within one state are related.

N -dimensional vector x is defined as the probabilities that word used in different n states, $x_i = \langle p_{i,j} : j \in (2, n+1) \rangle$, where $p_{i,j}$ defined as the probability that state j

containing $word_i$. Let w_j be a weight of $state_j$ for representing the favorite level of web user:

$$w_j = \sum_i favorite_{i,j} \quad (6)$$

As defined by formula 7, we employ an enhanced Cosine metric by incorporating the page weight.

$$Similarity(x, y) = \frac{\sum_{j=2}^{n+1} w_j \cdot x_j \cdot y_j}{\|x\|_2 \cdot \|y\|_2} \quad (7)$$

It measures similarity of two words according to the angle between them. Vectors pointing to similar directions are considered as representing similar concepts. By calculating the similarity of each pair of word, a similarity matrix is constructed, in which each vertex representing a word and each weight denoting the similarity between two words.

Step 3: Constructing MUI Based on Word Similarity Matrix

Given the similarity matrix of words, the clustering algorithm recursively divides clusters into child clusters, each of which represents a sibling node in the resulting MUI.

A threshold is used to decide whether two words are strongly related or not. At each partitioning step, edges with “weak” weights are removed from similarity matrix and the resulting connected components constitute sibling clusters. If two words are determined to be strongly related, they will be in the same cluster as belonging to same kind of interests of web user; otherwise, they will be in different clusters.

The recursive partitioning process stops when the current graph does not have any connected components after weak edges are removed or a new child cluster is not formed if the number of words in the cluster falls below a predetermined threshold.

MUI represents the user’s specific as well as general potential interests. The higher-level interests are more general, which represented by larger clusters of words. And the lower-level ones are more specific, which are represented by smaller clusters of expressions.

Step 4: Topic Mining According to MUI

As a cluster of words, the description of user’s interest in MUI is not easily digestible. So, we want to automatically summarize the topics of the leaf clusters, give a name to each topic. The name is used as a more specific representation of web user interests.

In practice, more general interests, in some sense, correspond to longer-term interests, while more specific interests correspond to shorter-term interests. In this way, a continuum of long-term to short-term interests of web user is mined dynamically.

4 Experimental Results

The study uses the following example to describe the execution of the dynamic semantic clustering algorithm. The sample data are described in Table 1. Using dynamic semantic clustering, these words in table 1 are represented by a MUI as shown in Figure 1.

Table 1. Sample data set

State	Content	Favorite
1	data mine cluster algorithm similarity function dynamic	0.9
2	HMM stochastic algorithm i.i.d. renewal process	0.9
3	uml oo java programming software engineer asp	0.8

The root of MUI represents the highest abstract of web user’s interest. It implies that the web user is interested in computer science and mathematics. It can be viewed as the long-term interests of web users, and motivate some other different specific interests. The leaves of MUI are specific interests of web user in short term. They might be changeable, but we can found that interests at lower level is a specialization of a higher level node, for example, ‘cluster’ and ‘similarity’ can be categorized as belonging to clustering algorithm. ‘HMM’ and ‘i.i.d.’ are belonging to Markov chain. At a higher abstraction, they are both in same cluster named as ‘Data mining and stochastic’. By this idea, the changeable of interest of web user can be tracked and represented dynamically.

To evaluate the effectiveness, we analyze the generated MUI in terms of dynamic performance, meaningfulness, and shape.

We categorized dynamic performance as ‘good’, ‘fair’, ‘bad’. A cluster is marked as ‘good’ when it can track the changeable of web user in leaf-clusters exactly. A cluster is marked as ‘fair’ when it can represent the changeable of web user in non-leaf clusters at a higher abstract level, otherwise, the cluster are marked as ‘bad’.

We categorized meaningfulness as ‘good’, ‘fair,’ or ‘bad’. A cluster is marked as ‘good’ when it has more than 2/5 of the words that are related. A cluster is marked as ‘bad’ when a leaf cluster has more than 15 words. ‘Fair’ leaf clusters are those that are neither good nor bad.

We categorized shape as ‘thin’, ‘medium,’ or ‘fat’. If a tree’s ABF value is 1, the tree is considered a ‘thin’ tree. If the ABF value of a tree is at least 10, the tree is considered a ‘fat’ tree. The rest are ‘medium’ trees.

Based on these valuation criteria, we analyze the MUI performance, as shown in table 2. The letter ‘D’, ‘M’, ‘U’, ‘ABF’ stands for dynamic performance, meaningfulness, user and average branching factor respectively. ‘G%’ means percent of good leaf clusters according to dynamic performance or meaningfulness. Table 2 illustrate that the dynamic performance (68%) and meaningfulness (62%) are good, and the shapes of MUI are mostly ‘medium’. Experimental results prove the effectiveness of the dynamic concept clustering algorithm.

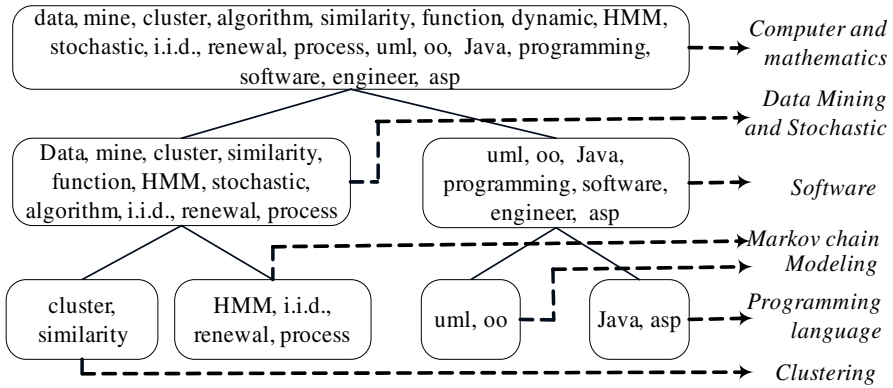


Fig. 1. Sample multilayer user interest

Table 2. Evaluation of MUI

User		U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	Sum
Leaf cluster		6	5	8	10	2	7	4	5	12	4	63
D	Good	6	3	2	5	2	7	3	4	7	4	43
	Fair		2	6	5			1	1	5		20
	Bad											
G%		100	60	33	50	100	100	75	80	58	100	68
M	Good	5	4	1	7	2	5	4	3	6	2	39
	Fair	1	1	7	3		2		2	6	2	24
	Bad											
G%		83	80	13	70	100	72	100	60	50	50	62
ABF		2	2.4	2	3.0	2.5	2.2	3	2.7	2.5	2.3	
Shape		M	M	M	M	M	M	M	M	M	M	

5 Conclusion

In this paper, we have proposed a semantic clustering approach for the dynamic continuum user interest. Markov use model tracks the preference page of web user. Some parameters, such as transition probability, mean stay time and favorite are defined in the model. The clustering algorithm generates a multilayer user interest. MUI represents the user's specific as well as general interests. The results of experiment show that the dynamic semantic clustering algorithm is effective and robust in tracking the dynamic user interest.

References

1. Perkowitz, M., Etzioni, O.: Towards adaptive web sites: conceptual framework and case study. *Artificial Intelligence*, 118(1-2) (2000) 245–275.
2. Sandhu, Y. Fu., Shi, M. K.: Clustering of web users based on access patterns. *LNAI1836*, (2000) 21–38.
3. Stadnyk, I., Kass, R.: Modeling user's interests in information filters, *Communication of the ACM*, 35(12) (1992) 49–50.
4. Belkin, N. J., Croft, W. B.: Information filtering and information retrieval: two sides of the same coin. *Communication of the ACM*, 35(12) (1992) 29–38.
5. Furnas, G. W., Landauer, T. K., Gomea, L. M., Dumais, S.T.: The vocabulary problem in human-system. *Communication of the ACM*, 30(11) (1987) 964–971.
6. Zhao, Y., Karypis, G.: Evaluations of hierarchical clustering algorithms for document datasets. *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, (2002) 515–524.
7. Duda, O. R., Hart, E. P., Stork, G. D.: *Pattern Classification*. Wiley, New York, (2001).
8. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (1999) 16–22.
9. Pelleg, D., Moore, A.: X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning*, (2000) 727–734.
10. I. Cadez, I., Gaffney, S., Smyth, P.: A general probabilistic framework for clustering individuals. Technical report, Univ. Calif., Irvine, (2000).
11. Smyth, P.: Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of 7th International Workshop AI and Statistics*, (1999) 299–304.
12. Ypma, A., Heskes, T.: Automatic Categorization of Web Pages and User Clustering with Mixtures of Hidden Markov Models. *LNAI 2703*, (2003) 35–49
13. Hafri, Y., Djeraba, C., Stanchev, P., Bachimont, B.: A Markovian Approach for Web User Profiling and Clustering. *LNAI 2637*, (2003) 191–202
14. Kominek, J., Kazman, R.: Accessing Multimedia through Concept Clustering. *CHI 97*, (1997) 22–27
15. Barbu, C., Simina, M.: A Probabilistic Information Filtering Using the Profile Dynamics. *IEEE International Conference on Systems, Man and Cybernetics*, (2003) 4595–4600
16. Billsus, D., Pazzani, M.J.: A Hybrid User Model for News Story Classification. *Proceedings of the 7th International Conference on User Modeling*, (1999) 99–108.

Building Interoperable Software Components Repository Based on MMF*

Bing Li, Keqing He, Jin Liu, Qiang Wang, Peng Liang, and Rong Peng

The State Key Laboratory of Software Engineering (SKLSE), Wuhan University
Wuhan, China, 430072
libingyk@public.wh.hb.cn

Abstract. Meta modeling is an effective approach to implement interoperability among many distributed and heterogeneous information sources. A MMF (Meta Model Framework) is a set of Meta objects and Meta modeling constructs to be used in the development a metamodel in the actual implementation of a registry. This paper proposes a common repository model based on MMF to ensure interoperability among heterogeneous software components repositories on the Web. The model will depict what aspects of model elements and constructs we will meet in metamodeling of software components repository. WHCRP (Wu-Han Component Repository Platform), a prototype system implemented based on MMF, is introduced in the paper.

Keywords: MMF (Meta Model framework), software component ontology, software component repository, registry model

1 Introduction

An information grid is a software infrastructure using the Grid technologies to achieve integrating, sharing and managing of some heterogeneous information resources scattered across disparate systems on the Web and to provide users or applications with information services on demand. Reusability based on Software component(SC) is an effective approach to implement large-scale software manufacturing production [1,2]. A prerequisite for software reuse is a repository that provides functions for classification, storage, management and retrieval of SC [3]. At present, lots of SC repositories, both public and private, already exist today, such as REBOOT [4], STARS [5] and JBCL [6], etc. However, due to difference motivation or difference establisher, those distributed repositories are autonomous and heterogeneous. That is to say, every repository has their own registry model, classification model and terms [6]. On one hand, the diversity of repository is necessary to maintain the ability of a SC repository to application for support concrete domain. On the other hand, it is difficult for a programmer who may be interested in some components stored in many

* This work was supported by the National Natural Science Foundation of China under Grant 60373086; Wuhan Science & Technique Key Project under Grant 20021002043; Open Foundation of SKLSE under Grant 03-03; Hubei Provincial Department of Education under Grant 2003A011.

different repositories to access and capture SCs in those repositories. Therefore, it is important to ensure interoperability among SC repositories so as to decrease user's burden of manipulation and increase reuse degree of SC. Software engineering in era of grid computing will depend on a SCs information grid that includes many interoperable SC repositories.

Although many organizations have been in charge of the development of registries standards which will facilitate interoperability such as RIG [7,8], a heterogeneity solution capable of deployment at web scale remains elusive. Using open standards including XML, SOAP, WSDL and UDDI, Web service is becoming a standardized way of integrating Web-based applications. Moreover, Web services use document style messages that offer more flexibility and more pervasiveness than other distributed object specification such as CORBA and DCOM[9]. On one hand, Web service enables reusable SCs to become products marketed on Internet. On the other hand, retrieving SCs is not limited to traditional SC repositories. Because different systems on Web, including traditional SC repositories, ebXML[10] and UDDI [11], describe their SCs with different manners. Registry model heterogeneity will increase rapidly. Then, it is more difficult for SC users to retrieve SCs. A feasible method is to found a common model that presents mapping to registry models of different system. Meta-models define the semantic of modeling elements and constructs that will be used to model the universe of discourse. As a standard draft to present a unified framework for metamodel interoperability MMF (Meta Model Framework) is developed by ISO/IEC/JTC1 in order to establish harmonization of the metamodels, which are developed independently and to reuse them widely across organizations[12].

Based on MMF and ontology for SCs, the paper presents a common SC repository model that achieves a registry framework according to MMF. The paper is organized as follows. In Section 2, we outline the MMF. In Section 3, we describe the SC repository model. In Section 4, we illuminate a prototype implementing the model. Finally, in Section 5, we give our concluding remarks.

2 Meta-model Framework

The metamodel framework family of standards consists of a core model of the meta-model framework and a series of metamodel framework, which are to be used in the development of a harmonized metamodel and materialization of the interoperation of existing registries or metamodels. The core metamodel is constructed on MOF (Meta Object Facility) [13] established by OMG(Object Management Group).

2.1 Meta object Facility

The heterogeneity in distributed computing environment, Web or grid, needs representation of meta information. In the area of software engineering, metamodel is widely used to describe various models. The MOF Model is referred to as a meta-metamodel which uses a common abstract syntax for defining metamodels in many typical technologies such as UML, XMI, CORBA, etc. Based on the traditional four layer(M3,M2,M1,M0) metadata architecture, MOF presents a "metadata architec-

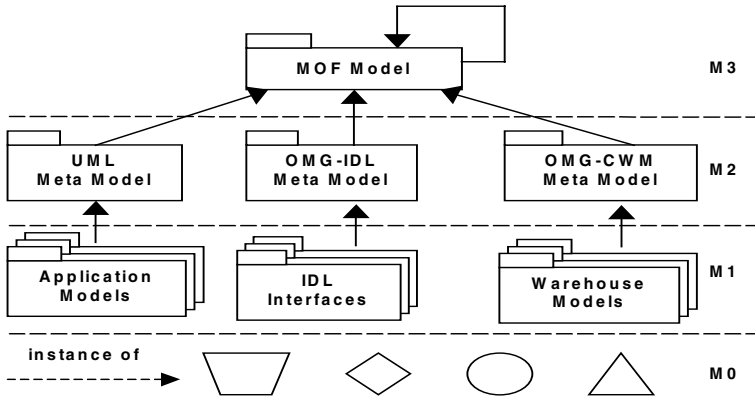


Fig. 1. MOF Metadata Architecture. M3: meta-meta model, M2: metamodel, M1: model, M0:object

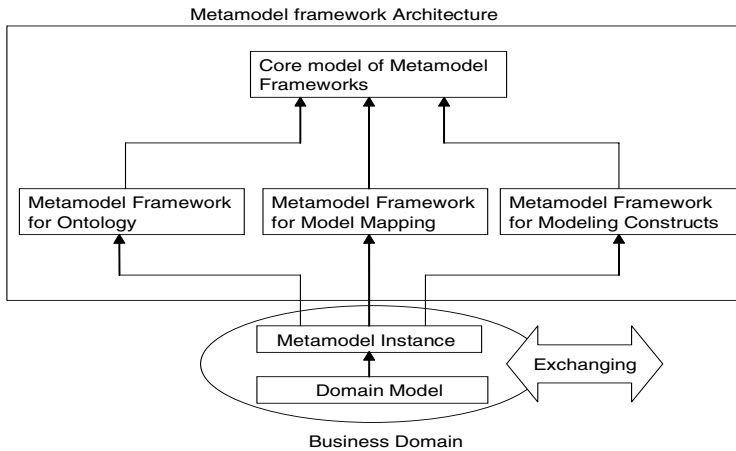


Fig. 2. Architecture of MMF (cited from [14])

ture”, illustrated in Figure 1, which defines UML Metamodel, CWM metamodel and other metamodels. Moreover, a mechanism based on MOF/XMI can support interoperability of model information among different platforms.

2.2 MMF (Meta Model Framework)

MOF is also the foundation of Meta Model Framework. Currently, MMF architecture consists of four parts: a core metamodel, metamodel framework for ontology, metamodel framework for mapping and metamodel framework for model constructs (See figure 2). However, other useful metamodel frameworks should be proposed in the future [12].

In this family of standards, the core model could govern every metamodel framework and they should be developed inheriting concepts and constructs of metamodel frameworks of the core model. The core model should be formulated by inheriting both MOF native metamodels and MDR (ISO/IEC 11179-3) Metamodel, accordingly all of metamodel frameworks have to follow the metamodel concept and basic meta objects of MOF and MDR[14]. The metamodel frameworks in this family of standards should be formulated on UML and MOF.

3 A Common Model of Software Components Repository

Based on MMF, we present a common model of SCs repository. Figure 3 illustrates its architecture.

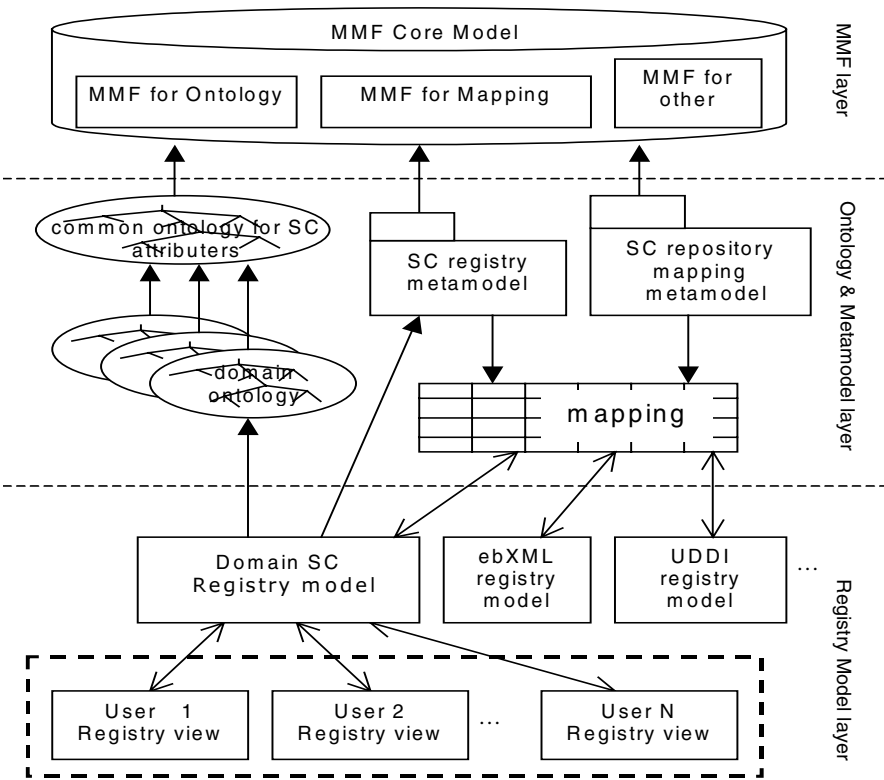


Fig. 3. A Common Model of Software Components Repository

The model consists of three layers. The top layer is MMF including its specifications which ensure the interoperability among heterogeneous software components repositories. The middle layer is Ontology & Metamodel layer, including SC attribute ontologies, SC registry metamodel and SC repository mapping metamodel. The lowest level is registry model layer.

3.1 Software Component Attribute Ontology

Ontologies provide a shared and consistent understanding of data (and, in some cases, services and processes) that exists within in specific Universe of Discourse(UOD), and how to facilitate communication between people and information systems [15]. Studies have shown that ontologies do great help in many aspects in the information modeling and integrating, such as overcoming semantic heterogeneity among Web-based information systems [16].

Today, various consortia or organization defined schemas with their own manners in the term of the SC attributes. Though inheriting MMF, a common ontology for SC attributes unifies the different concepts and eliminates conflicts existing among registry repository. Furthermore, many typical domain ontologies of SCs could be established for different application environment. Those ontologies embody SCs as an explicit set of concepts, their definitions, attributes and inter-relationships in order to support identifying, classifying and consistency checking in registry.

3.2 Software Component Registry Metamodel

MMF is the specification on metamodeling, which is independent from concrete application domains. However, as a application in the area of information classification on Web, SC registry depend on strength metamodel to ensure coherence among SC registry mechanisms in different domains. Governed by MMF, especially MMF for ontology [15], SC registry metamodel not only inherit MMF elements so that SC registry for different applications is easy to understand each other, but also obtain metamodel interoperability between SC registry and non-SC registry. Figure 4 shows a “light-weight” SC registry metamodel we develop.

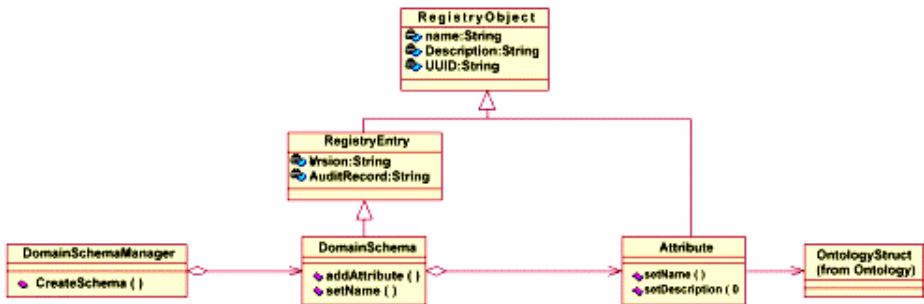


Fig. 4. A SC registry metamodel

3.3 Registry Model Layer

The registry model layer of Common Model of SCs Repository includes various registry models. Some registry models developed for special domain are usually stored in special SCs repository. Registry contents of those SCs are different from each other as well as number of registry items. For example, SCs in domain of mobile phone games only have 10 registry items. In contract, SCs in domain of GIS (Geographical Infor-

mation Systems), e-bank or Web Service might have 50-60 registry items. In addition, some registry models follow the existed specifications, such as ebXML registry model and UDDI registry model for Web Service. Those registry models are developed not only for SCs but more business objects. Relying on registry metamodel and repository mapping model, it is possible for those registry models achieve interoperability.

4 Implement of the Common SC Repository Based on MMF

Based on above models, WHCRP (WuHan Component Repository Platform), a SCs repository prototype is implemented. Figure 5 shows architecture of WHCRP.

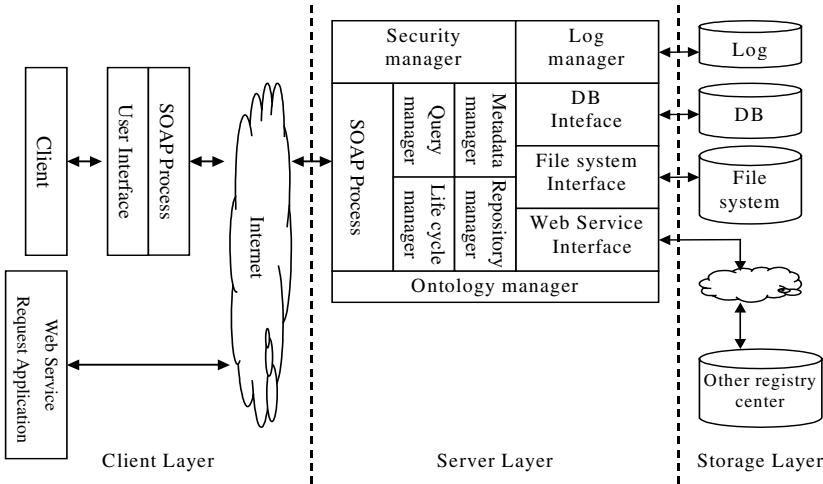


Fig. 5. Architecture of WHCRP

Design and development of the prototype use J2EE (Java 2 Platform, Enterprise Edition) technologies. Functions of each part in the system are explained as follows:

- **Client Layer.** Client layer consists of user interface and SOAP process. User interface provides access to repository through two views: a customize interface for users and a general Web services API. Because metamodel presents mapping for different registry models, a customize interface means users could adopt a habitual method to register and retrieve SCs. Interaction between client and server uses the Simple Object Access Protocol (SOAP) over HTTP.
- **Sever Layer.** Main functions of the layer include ontology management, metadata management, query processing, life-cycle management and repository management. There are three kinds of interface adapting to various access storage systems: database, file system and other registry center using Web service also.
- **Storage Layer.** In our system, there are two kinds of information: meta information of SC registry stored into a database and files of SCs in file system.

5 Conclusion

The objective presented in this paper is relevant with regard to resource interoperability in information grid. Nowadays, reusable Software components on Web are important resource for software engineering. Interoperability among many distributed and heterogeneous SC repositories is regard as a key factor of successful software reuse. MMF (Meta Model Framework) is the framework standard developed by ISO to ensure metamodel interoperability. In this paper, we provide a common SC repository model based on MMF. Combining meta-modeling with ontology, the model enables various registry models, even self-defined registry model to understand each other. By the model, we've built a prototype of SC repository, named WHCRP (WuHan Component Repository Platform), which uses ontology and Web services to create a software component repository platform that offers user-centric support for SCs management and retrieval. The target of our prototype is to achieve transparent access to other SC repositories adopting heterogeneous registry from one repository. Because MMF is yet a working draft, we will follow its advance and improve our repository model.

References

1. Alessandro Orso, Mary Jean Harrold, David Rosenblum. Component Metadata for Software Engineering Tasks. Advanced Information Systems Engineering. Proceedings of the 13th International Conference of Advanced Information Systems Engineering (CaiSE2001), 2001
2. Heiko Hahn, Klaus Turowski. Drivers and inhibitors to the development of a software component industry. Proceedings of the 29th EUROMICRO Conference 2003, pp: 128-139, 2003
3. Morel JM, Faget J. The REBOOT Environment. Proceedings of the 2nd International Workshop on Software Reusability Advances in Software. Lucca: IEEE Computer Society Press, 1993. 80-88
4. STARS. Asset Library Open Architecture Framework Version 1.2. Informal Technical Report, START-TC_0404/001/02, August, 1992
5. Keqin Li, Lifeng Guo, Hong Mei, Fuqing Yang. An Overview of JB (Jade Bird) Component Library System JBCL. Proceedings of the 24th International Conference TOOLS ASIA, Beijing, 1997, pp. 261-267
6. Yang Fu-qing, Shao Wei-zhong, Mei Hong. The design and implementation of an object-oriented CASE environment in jade bird 2 system. Science in China (Series A), 1995, 25(5):533-542
7. Reuse Interoperability Group (RIG). Basic Interoperability Data Model. Technical Report, RPS0001, Reuse Library Interoperability Group. 1993
8. Reuse Library Interoperability Group. RIG Uniform Data Model for Reuse Libraries (UDM). 1994
9. Jen-Yao Chung, Kwei-Jay Lin, Richard G. Mathieu. Guest Editors' Introduction: Web Services Computing: Advancing Software Interoperability. IEEE Computer, 2003, 36(10): 35-37
10. OASIS/ebXML. Registry Information Model v2.0. Approved Committee Specification, <http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebxml.pdf>

11. Universal Description, Discovery and Integration (UDDI). UDDI specifications.
<http://www.uddi.org/specification.html>
12. International Standard Organisation (ISO). Information Technology – Framework for Metamodel Interoperability – Reference Model (Technical report). ISO/IEC 19763-1, Project Editor: Hajime Horiuchi,
<http://metadata-stds.org/metadata-stds/Document-library/Meeting-reports/SC32WG2/2003-10-Melbourne/WG2-MEL-047-WD19763-01-MMF-RM.doc>
13. Object Management Group (OMG). MOF 1.4 specification.
<http://www.omg.org/cgi-bin/apps/doc?formal/02-04-03.pdf>
14. International Standard Organisation (ISO). Information Technology – Framework for Metamodel Interoperability – Core Model (Technical report). ISO/IEC 19763-2, Project Editor: Masaharu Obayashi,
<http://metadata-stds.org/metadata-stds/Document-library/Meeting-reports/SC32WG2/2003-10-Melbourne/WG2-MEL-018-WD19763-2-MMFCore031020.doc>
15. International Standard Organisation (ISO). Information Technology – Framework for Metamodel Interoperability – Metamodel Framework for Ontology (Technical report), ISO/IEC 19763-3, Project Editor: KeQing He (SKLSE, China)
16. Guarino N. and Welty C, A Formal Ontology of Properties, In Proceedings of Knowledge Engineering and Knowledge Management: Methods, Models and Tools. Proceedings of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW2000), Springer Verlag: 97-112, 2000

An Approach for Constructing Software Component Repository in Grid Environments*

Dehui Du, Shi Ying, Keqing He, and Yangfan He

State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China
School of Computer, Wuhan University, Wuhan, China
d_dehui@hotmail.com

Abstract. Software component repository is practical and efficient for reusers to develop software based on components. To share component resources in the grid environments, there should be a software component repository, which can realize the sharing of software components resources and related services. In this thesis a framework to construct software component repository by web service and grid technology is proposed. In the framework, an improved reuse-oriented faceted classification to accomplish scientifically classifying and managing software components is included. Utilizing the feature of supporting external taxonomies in UDDI2.0, the component classification is integrated into UDDI in the form of tModel and the taxonomy validation service associated with the classification is also given. Finally, we give a prototype system based on this framework and discuss the future work.

1 Introduction

Component-based software engineering (CBSE) has extensive and deep impact on the development of software [1]. Software component repository (SCR) is the practical technique for CBSE. However, the existing SCR can't satisfy the needs of sharing resources in the grid environments. Nowadays, Grid technologies [2, 3] are getting more popular and have been applied to various computational fields. The Grid infrastructure can support the sharing and coordinated use of diverse resources in dynamic and distributed virtual organizations [3]. To accomplish the purpose that reusers can find available components in the grid environments, we should construct a SCR by grid technologies. Because web service is a practical and effective technology which realizes the sharing of resources and related services in the grid environments, in the form of web-based services to manage and share components resources effectively is a new solution to this problem. This paper proposes a framework to construct the SCR in grid environments, which adopts the improved faceted classification for components and utilizes the feature of supporting external taxonomies in UDDI2.0 [4] to integrate the classification into UDDI.

* This research was supported by the Hubei Outstanding Young Scientist's Foundation under Grant 2003ABB004, National Science Foundation of China under grant 60373086; Wuhan Science & Technique Key Project under Grant 20021002043; the Science Foundation of Hubei under grant 2002ABB037; Open Foundation of SKLSE under Grant 03-03;

Besides, Reusers can utilize the specific index engine to index software components in the SCR based on UDDI. This kind of SCR can provide web-based services for reusers in grid environments. It is the most pronounced specialty that the framework has.

2 Related Work

Nowadays, some organizations are researching the reusable component repositories. Among these repositories, there are some typical example, such as RIG which is launched by Reuse Library Interoperability Group [5][6] and NATO etc. There is also ALOFAF model of STARS (Software Technology for Adaptable & Reliable System) [7]. STARS is launched by the American Military and supported by CMU SEI/MITRE. The aim of STARS is that merging the methods and technologies of reuse into software engineering. This kind of repository doesn't deal with the application situations in the grid environments so it can't satisfy the demands of reusers on the Internet to share the component resources and related services with SCR. Moreover we find their classifications are too simple to provide reuse-oriented information. We should improve the classifications to meet the needs of high efficiency of index components in the grid environments.

Another problem is that there also must be a specific index engine to index software components effectively. To solve the problems of the existing software component repository, a new framework for constructing a SCR in the grid environments is proposed in the following.

3 Reuse-Oriented Faceted Classification

In 1987, DR.Ruben Prieto-Diaz proposed the software component classification, which is based on faceted classification [8]. Its faceted classification scheme has six facets such as: Function facet, Media facet, Object facet, System type facet, Functional area facet, Setting facet. It also establishes the scheme of classes according to these facets. For example, the functional facet, its scheme of classes is: add, compare, compress, build etc. However, the original component classification has evident limitations:

- The classification is too simple to satisfy the needs of reusers, i.e there are not enough reuse-oriented facets and many facets mean nothing for reuse in essence.
- Component information is not enough for reuser to find and reuse components.
- Very difficult to define classification scheme and term lists.

In order to solve these problems much better, the thesis focuses on analysis the useful information for the reuse process and proposes a reuse-oriented faceted classification (ROFC). In the ROFC, we extend some reuse-oriented facets:

- Facets that describe the domain expert's taxonomy technology;
- Facets that describe the usage of software components in application domain;
- Facets that describe the software component model;
- Facets that describe the development circumstance of software components;

Then, the new classification scheme is as follows:

1. Basic facet

- 1.1 Management Attribute
 - 1.1.1 Provider Information
 - 1.1.2 Product Information
 - 1.1.3 Providing Method
 - 1.1.4 Additional Information
- 1.2 Technical Attribute
 - 1.2.1 Applicable Domain
 - 1.2.2 Running Environment
 - 1.2.3 Development Environment
- 1.3 Interface Attribute
 - 1.3.1 Name
 - 1.3.2 Functional Declaration
 - 1.3.3 Parameter
 - 1.3.4 Return Data
 - 1.3.5 Pre-condition
 - 1.3.6 Post-condition

2. Domain Facet

- 1.4 Business Attribute
- 1.5 Function Attribute
 - 1.5.1 Function
 - 1.5.2 Performance
 - 1.5.3 Additional Information
 - 1.5.4 Extendable Function
- 1.6 Structural Attribute
 - 1.6.1 Construction Information
 - 1.6.2 Information of Other Dependent Software Component

All these facets are reuse-oriented and got by our practical research work. Because the scheme of classes is very clear and precise on each facet, components can be located in the facet which is just cared by the reusers. For example, when the reuser retrieval a certain functional software components, they only need to input keywords of the function. The components can be located in the functional attribute of domain facet, then, according to the function, performance, additional information, extendable function of component, the reuser can select the proper one from the repository. These facets are very meaningful for reuse process in essence. This process of retrieval is completely reuse-oriented. We have also made an experiment of comparing with the original faceted classification. Through the analysis and compare, the advantages of the ROFC are as follows:

- Describe all kinds of detailed information to provide sufficient support for choosing components.
- Each facet can be combined flexibly, which gives all kinds of subject information related to components.
- Structure of classification scheme is easy to modify and fit for dynamic update. So the improved faceted classification has practicality and efficiency for the SCR in the grid environments.

4 Construction of the SCR in Grid Environments

4.1 UDDI2.0 and Its Support for the External Taxonomy

Universal Description, Discovery and Integration (UDDI), is the specification of information registry of distributed web service. It provides an interoperational and foundational infrastructure for web service-based software development. Through UDDI, people can publish and find the information of some company and its web-based service. And according to the information saved in UDDI, these services can be invoked by a unified method.

UDDI2.0 has the characteristic of supporting the external taxonomies and provides the standard APIs to realize the taxonomy validation service. We can utilize this mechanism to extend the capacity of UDDI operator, which makes the operator support the ROFC and integrate it into the UDDI Registry. Through the analysis above, we know there are three problems in the process of construction the SCR: (1) How to extend the taxonomy scheme of UDDI, which can make UDDI support the management of software components. (2) Give the tModel, which describes the ROFC for software components and registry the tModel into UDDI Registry. (3) How to provide the taxonomy validation service associated with the ROFC to assure UDDI operators validate the registry information according to the classification.

4.2 Registry the ROFC into UDDI Registry

Because there is no component classification among the three built-in common classifications in UDDI, the thesis takes full advantage of the characteristic of UDDI2.0 for supporting the external classifications and integrates the ROFC in preceding context into UDDI Registry. The essence of this action is to add a new classification node to UDDI classification tree and the classification node is used to describe the ROFC.

The result is that UDDI can utilize the ROFC to classify and manage software components saved in UDDI Registry. In addition, the providers of component classification also need publish the standard APIs to accomplish taxonomy validation service. The service will validate the registry information to assure the saved information conform to the classification. The work model of the extended SCR is as follows:

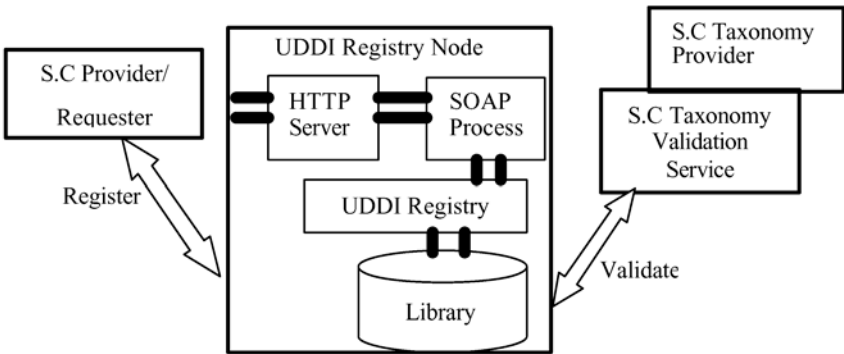


Fig. 1. Work Model of the SCR Based on UDDI

In the above figure, S.C Taxonomy Provider represents the providers of component classifications, which provide the ROFC. S.C Taxonomy Validation Service represents the taxonomy validation service of software components. S.C Provider/Requester represents the person who registry or reuse components. The three parts are the extended parts according to the support for the external taxonomy in UDDI2.0. The rest parts of the figure are not modified, just the same as the responding parts in UDDI. Its workflow is as follows: Firstly, the providers registry the technical information about the ROFC into UDDI Registry. Secondly, registry the taxonomy validation service into UDDI Registry. Thirdly, the providers of software components registry the information about themselves and utilize the ROFC to manage the information in UDDI Registry. Finally, UDDI invokes the taxonomy validation service to validate registry information.

4.3 tModel for the ROFC

According to the special characteristic of software components, a kind of tModel should be created for describing the ROFC and be added to the UDDI tModel tree to accomplish adding a new component classification in the UDDI classification scheme. To understand easily, the thesis gives a simple example, which illustrates the registration of the classification that needs to be checked in UDDI Registry. The detailed description of the faceted classification is as follows:

```
<tModel authorizedName="..." operator="..."
tModelKey="uuid:22222222-3333-4444-5555-666666666666">
<name> Software Component Faceted Classification
</name>
<description xml:lang="en">
Extendable taxonomy used to categorize software compo-
nent.
</description>
<overviewDoc>
<description xml:lang="en">Taxonomy of Software Compo-
nent categorization. Only listed values can be refer-
enced. Offered only to licensed members.
</description>
<overviewURL>http://www.SKLSE.org/software component
faceted classification.html </overviewURL>
</overviewDoc>
<categoryBag>
<keyedReference tModelKey="uuid:C1ACF26D-9672-4404-
9D70-39B756E62AB4" keyName="uddi-org:types" key-
Value="categorization"/>
```

```

<keyedReference
tModelKey="uuid:C1ACF26D-9672-4404-9D70-
39B756E62AB4"keyName="uddi-org:types" key-
Value="checked"/>

<keyedReference tModelKey="uuid:C1ACF26D-9672-4404-
9D70-39B756E62AB4"

keyName="uddi-org:types" keyValue="unvalidatable"/>

</categoryBag></tModel>

```

The tModelKey of the tModel is produced by UDDI Registry Center randomly. Besides, name, descriptive information and overview document of the tModel provide the detailed information about the ROFC. The tModel is marked by unvalidatable in keyedReference, which enunciates the ROFC is unavailable before the registry process of the classification is completed. Only after the external classification is integrated in UDDI Registry and the key value is modified with “validatable”, the classification is available.

For the external component classification, when the tModelKey associated with the classification is referred by the keyedReference, the value of keyValue will be validated by the corresponding taxonomy validation service. Only the validated legal data of keyValue can be saved with “checked”. If the data can’t pass the validation, the validation service will mark the data with “unchecked”. Then how the providers provide the taxonomy validation service?

4.4 Taxonomy Validation Service Provided by the Provider of the ROFC

Whenever UDDI operator invoke the APIs which are used to save registry information, such as, save_business, save_service or save_tModel[9], all the information of categoryBag in its parameter set will be validated. The process of validation is controlled completely by the third party entity. The third party entity must provide a web service which has the same style as UDDI (for example, using the SOAP1.1 based on HTTP as the transmission mechanism). Meanwhile, the third party must publish a simple function named of validate_values to accomplish the taxonomy validation service. In order to validate the information in the SCR based on UDDI, the providers of the ROFC must provide the API for the taxonomy validation service. The following is the simple description of the taxonomy validation service:

```

<validate_values generic="2.0" xmlns=" rn:uddi-
org:api_v2">

<bussinessEntity/>Software Component Pro-
vider</bussinessEntity>

<bussinessService/>Software Component Ser-
vice</bussinessService>

<tModel/> Software Component Faceted Classification
</tModel></validate-values>

```

The businessEntity declares the validated providers of software components. BussinessService declares the web service which is provided by the providers of components, tModel declares the ROFC for components. When UDDI operator invokes the validate_values, it will pass a bussinessEntity or bussinessService or tModel as the only parameter to the function according to the practical need. The parameter just is the one which is passed to save_bussiness or save_service or save_tModel, that is, it is the parameter passed when the registry information is saved. For example, if tModelKey according to the ROFC is passed to save_tModel, in order to invoke the taxonomy validation service, when the validate_values is invoked, the tModel marked by the tModelKey will be in the parameter set of validate_values. UDDI operator will validate the information according to the specification of the tModel. All this makes the information in UDDI Registry credible in some sense.

4.5 Implementation

About the implementation of the framework, we are developing the prototype system of the framework. The prototype system of the specific index engine has finished. The prototype system of the SCR is under development. It can share software components resources by web service in the grid environments, and provides the convenient support for reusers to search components. Due to the limits of this thesis, we will not give the details of the specific index engine.

We only illustrate reusers search for components by keyword-based search provided by the index engine. Reusers submit the search request by entering the keyword. As shown in the Figure 2, reusers enter the “EJB” as the search keyword in the input box, and then the specific index engine returns the search results on the right web page. The results show that the index engine finds thirty-six candidate components. Reusers can click the links of any candidate to check the components in details.

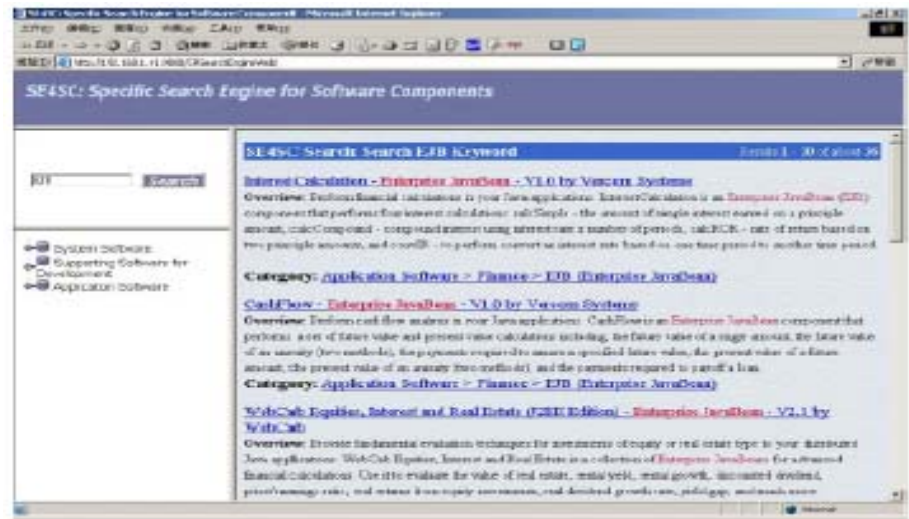


Fig. 2. Keyword-based search by index engine

5 Conclusion

The thesis analyses how to construct the SCR based on UDDI to accomplish the registration, management, index of software components in the form of web service in grid environments. To improve the efficiency of index, we propose the ROFC and integrate it into UDDI2.0. We also give the taxonomy validation service according to the ROFC. The framework provides a new approach to solve the problems existing in CBSE, which makes reusers utilize component resources on the Internet through web service in grid environments. And the framework has some expansibility, so the providers of the external classification can use the similar approach to integrate component classification to UDDI Registry.

The future research work is how to optimize the prototype of the framework to make it friendly for users and improve the reuse-oriented faceted component classification further to raise the efficiency of index software components. The scheme of classes of facets that are reuse-oriented should be optimized and the index engine should be integrated in the SCR seamlessly.

References

1. Szyperski C., *Component Software*, Addison Wesley, 1999
2. I. Foster, C. Kesselman, J. Nick and S. Tuecke, "Grid Services for Distributed System Integration", *Computer*, 35(6), 2002
3. I. Foster, C. Kesselman, J. Nick and S. Tuecke "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Proceedings of the 5th Global Grid Forum Workshop (*GGF5*), Edinburgh, Scotland, July 2002.
4. Providing a Taxonomy for Use in UDDI Version 2.0, OASIS Open, <http://www.oasis-open.org/committees/uddi-pec/doc/tcspecs.htm#uddiv2>, 2002
5. Reuse Library Interoperability Group, "RIG Basic Interoperability Data Model (BIDM)", RPS-0001, April 1993.
6. Reuse Library Interoperability Group, "RIG Uniform Data Model for Reuse Libraries (UDM)", RPS-0002, January 1994
7. STARS, "Asset Library Open Architecture Framework Version 1.2", Informal Technical Report, STARS-TC-04041/001/02, August, 1992
8. Johes.G, Prieto-Diaz.R, "Building and Managing Software Libraries", Computer Software and Applications Conference, 1988. COMPSAC 88. Proceedings, Twelfth International
9. UDDI Vision 2.0 API Specification, UDDI Version 2.0, OASIS Open, <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.html>, November 2002.

Agent-Based Modeling for Virtual Organizations in Grid

Yuqing Zhai, Yuzhong Qu, and Zhiqiang Gao

Department of Computer Science and Engineering
Southeast University, Nanjing, P. R. China, 210096
{yqzhai, yzqu, zqgao}@seu.edu.cn

Abstract. The Grid is an emerging technology for enabling resource sharing and coordinated problem solving in dynamic multi-institutional virtual organizations. In this paper, an agent-based modeling approach for virtual organization in Grid is proposed. An agent-based framework for virtual organizations is presented, and an ontology-based matchmaking mechanism in the framework is proposed. The agent-based approach can meet the requirements of scalable, dynamic, autonomous architecture of virtual organization, and fit in with the development of Semantic Grid.

Keywords: Virtual Organization, Grid, Agent, Ontology

1 Introduction

The Grid is an emerging technology for enabling resource sharing and coordinated problem solving in dynamic multi-institutional virtual organizations [1]. The core of the grid is sharing of resources and coordinated problem solving in virtual organizations. A virtual organization (VO) is a set of individuals and/or institutions with some common purpose or interest and that need to share their resources to further their objectives. Different individuals or institutions may have different usage policies and pose different requirements on acceptable requests. These shared resources are typically computers, data, software, expertise, sensors, and instruments. A virtual organization has usually dynamical members and is across multiple institutions. Now there are many challenges in the construction and evolution of a virtual organization, such as coordination and safety assurance mechanism under autonomous conditions, system usability and flexibility in the heterogeneous environment. There have been some researches on member management, resource sharing policy and trust model [2-6]. But the (formal) models on virtual organizations need to be studied further, for example, expressing and performing of policies on member management and resource usage, inter-operation and coordination among virtual organizations, trust models in virtual organizations. How can we model virtual organization and realize the sharing of resources and coordinated problem solving in a virtual organization? We can make full use of agent technology originated in the field of distributed artificial intelligence (DAI)[7].

A fundamental property of an agent is autonomy: an agent operates without direct interference by humans or other systems, and has control over its behavior and its

internal state. The concept of an intelligent agent extends this definition by the capability of acting flexibly, whereby the notion of flexibility comprises three characteristics: reactivity (agents perceive their environment and react timely and appropriately to changes within this environment), pro-activeness (agents do not only react to observed changes within their environment, but are capable of taking the initiative in a goal-directed fashion), social ability (agents interact with other agents (and possibly humans) by exchanging information formulated in an agreed-up communication language). Moreover, the notion of social abilities comprises complex patterns of behavior based on communication protocols, e.g. for the purpose of negotiation. This concept of intelligent agents is perfectly suitable for the domain of coordinated problem solving. First, an agent has some kind of knowledge of the problem to be solved and its environment (e.g., other agents), and is capable of negotiation. Second, it is able to quickly react to changes within its environment, e.g. a machine breakdown. And third, agents are pro-active, allowing them to improve their planning schedule while no other service request are issued.

A number of initiatives to apply agents in grids have been initiated in recent years. Manola and Thompson [8] present an overview of different perspectives to grid environments and describe DARPA's Control of Agent-Based Systems (CoABS) agent grid where agent technology is expected to help to provide more reliable, scalable, survivable, evolvable, adaptable systems, and help to solve data blizzard and information starvation problems. A good example of an agent grid is presented by Rana and Walker [9], where an agent based approach to integrate services and resources for establishing multi-disciplinary problem solving environments is described, in which specialized agents contain behavioral rules, and can modify these rules based on their interaction with other agents and with the environment in which they operate.

In this paper, we focus on applying agent technology in modeling virtual organizations in order to provide more feasible services. In a virtual organization, we assume that all resources in the virtual organization be monitored by agents and one agent can monitor several resources. That means each institution (as a member of a VO) has its agent. So a virtual organization is composed of the agents who monitor and control the usage policies of the related resources, and can be considered as brokers of institutions and/or individuals. Further, a virtual organization can be one member of another virtual organization. We name the virtual organization containing virtual organizations as a complex virtual organization. If a virtual organization does not contain any other virtual organization, we name the virtual organization as a simple virtual organization.

The fundamental task of a virtual organization is just to provide feasible services. How to connect services provided by a virtual organization with requests of an end user? The basic idea is matchmaking. There are two main kinds of matchmaking methods: attribute-based and semantic-based. We focus on semantic-based matchmaking in this paper. In order to make semantic-based matchmaking come true, we propose OWL (Web Ontology Language)-based service description and ontology-based matchmaking. An ontology is a specification of a conceptualization [10], a formal model of a shared understanding within a domain. Ontology is considered as a key technology for Knowledge Management largely for their promise of bringing a

consensus in the way a particular area of expertise is described. The OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web [11]. Agent-based model is suitable for scalable, autonomous environments as Internet. OWL-based description is adaptive to semantic matchmaking for services.

The rest of the paper is organized as follows. In Section 2, we give the framework of agent-based virtual organizations. In the framework, we propose OWL-based specifications on services provided by agents in virtual organizations. In Section 3, we discuss how to match the requirements of a user's request for services provided by virtual organizations from the view of semantic. Finally, we conclude our paper in Section.

2 An Agent-Based Framework for a Virtual Organization

A virtual organization (VO) is a set of individuals and/or institutions with some common purpose or interest and that need to share their resources to further their objectives. Different individuals or institutions may have different usage policies and pose different requirements on acceptable requests. In order to provide efficient services for a service requestor and coordinate among resources in a virtual organization, we assume each institution (as a member of VO) in the virtual organization be bound to an agent whose responsibilities are monitoring, communication. And the concept of master agent is proposed to realize the coordination of problem solving and sharing of common knowledge and information in a virtual organization. There is only a master agent in a virtual organization.

We divide virtual organizations into two sorts: simple virtual organization and complex virtual organization. In a simple virtual organization, all agents but the master agent monitor actual resources. And an agent may belong to two or more virtual organization as the case in a multi-agent system. In a complex virtual organization, some agents are master agents of other virtual organizations (simple or complex), that is, a complex virtual organization contains virtual organization.

An agent monitoring actual resources can be modeled as Fig.1.

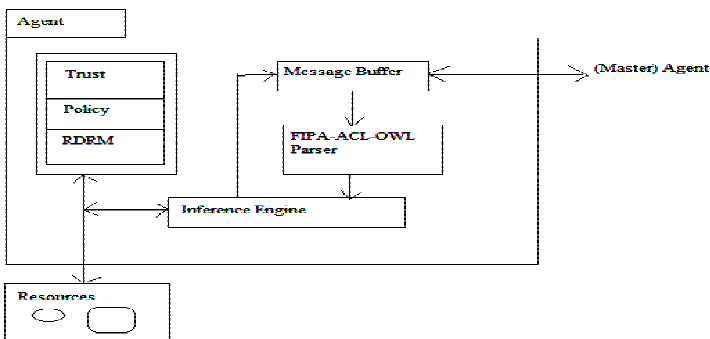


Fig. 1. The architecture of an agent monitoring actual resources

In Fig 1, the module “RDRM” (Resource Digital Rights Management) is used to realize the access control to the related resources. The essence of the access control is just a kind of digital rights management, that is, the agent issues some rights, e.g. run or read, to a service requestor (resource consumer) when the requestor satisfies some pre-conditions, e.g. paying some money. The resources here may be CPU, memory, instruments, programs, or web services. The owner of resources would describe the restrictions on usages of the resources, such as using after paying. At present techniques on Digital Rights Management are growing up. The core of Digital Rights Management is Digital Rights Expression Language. We have proposed an ontology-based digital rights expression language (OREL) in a China High Technique Project (863). We would use the ontology-based description method in module RDRM.

The module “Policy” is just used to describe the rules on resource usage, such as operation strategy and security policy. Here we just consider VO resource policy [3]. VO resource policy describes the virtual organization’s rules for the behavior of its member resources. This type of policy is useful for setting rules that pertain to particular resources within the VO rather than across the entire organization. And these rules would be described in OWL Rules Language [12].

The module “Trust” is used to depict the axioms for trust relations among agents in a virtual organization or agents over several virtual organizations, and trust values in order to decide that an access to a specific resource in a virtual organization is whether allowed or not, from the security trust relations and trust values. We could describe the axioms for the trust relations in OWL Language [11]. Module “RDRM”, “Policy” and “Trust” should be considered as a whole in the inference engine.

The module “FIPA-ACL-OWL Parser” is used to interpret the messages and contents in the message from other agents, and pass the results to the inference engine. Here we assume that the agents are FIPA-compliant agents, and the messages and the contents in the messages are described in OWL content language for FIPA ACL [13].

The module “Inference Engine” is just used as an inference engine. It is the core of the agent monitoring actual resources. The engine receives information about the requirements on the related resource usage from “FIPA-ACL-OWL Parser”, and information about the digital rights expression, resource usage policy and trust relations and axioms from “RDRM”, “Policy” and “Trust”. Then it makes matching between requirements and resource usage descriptions. Finally it gives a response message. The basic inference mechanism is a mechanism that combines rule-based inference mechanism with ontology-based inference mechanism.

A simple virtual organization can be modeled in Fig.2.

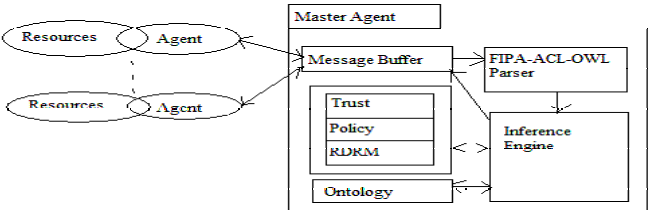


Fig. 2. The architecture for a simple virtual organization

In Fig 2, the structure of the Master Agent is just similar to that of an agent monitoring actual resources except that the master agent has not linked resources, and the resources digital rights expression, policy, trust is common to the whole virtual organization, and there is an ontology for common domain-specific knowledge representation in the virtual organization. And the inference engine includes more functions, such as dividing and conquering of a problem, accessing the related ontology.

A complex virtual organization can be modeled in Fig 3.

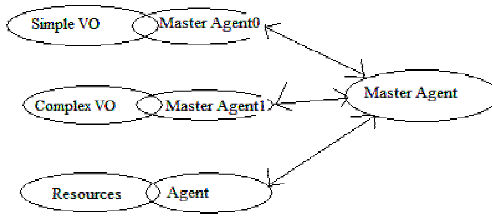


Fig. 3. Architecture for a complex virtual organization

Note that there may be multi-level ontologies in a complex virtual organization.

3 Ontology-Based Matchmaking in Virtual Organization

In a virtual organization, the agents have different constraints that can only be satisfied by certain types of resources with specific capabilities, that is, each agent has its own capabilities. Before a resource (or a set of resources) that is bound to an agent can be allocated to run an application, the requestor of the application must select resources appropriate to the requirements of his application. At the time, matching requirements with agents' capabilities should be made. Traditional matching, as exemplified by the Condor Matchmaker [14] or Portable Batch System [15], is done based on symmetric, attribute-based matching. In these systems, the values of attributes advertised by resources are compared with those required by jobs. For the comparison to be meaningful and effective, the resource providers and consumers have to agree upon attribute names and values. The exact matching and coordination between providers and consumers make such systems inflexible and difficult to extend to new characteristics or concepts. Moreover, in a heterogeneous multi-institutional environment, it is difficult to enforce the syntax and semantics of resource descriptions. Here we introduce an ontology-based matchmaking method to realize semantic matchmaking.

An ontology-based matchmaker needs at least three components: the ontologies (capturing the domain model and vocabulary for expressing resource advertisements and job requests), domain background knowledge (capturing additional knowledge about the domain), and matchmaking rules (defining when a resource matches a job description). In the agent-based virtual organization framework, ontologies and domain background knowledge (as a part of ontology, and expressed in OWL) are included in the master agent, which can describe the object and object relations in the

specification of resource, resource request, resource digital rights expression, resource usage policies, trust and trust relations. And matchmaking rules are embedded in the inference engine in an (master) agent.

The matchmaking process is as following:

- a) An user describes a resource application requirements in the terms in the ontology in a virtual organization and issues the requirements
- b) An agent in the virtual organization receives a message
- c) The agent's FIPA-ACL-OWL parser parses the type and the content of the message
- d) The parser passes the parsing result to the inference engine
- e) The inference engine makes matching between his own capabilities, constraints and the requirements according to the digital rights expression, usage policy and trust rules
- f) If the inference engine draws a conclusion that the agent can satisfy the requirements, it responses a confirm message
- g) If the agent cannot satisfy the requirements, it would forward the received message to the master agent or his trusting agent according to the matchmaking rules and the policy

When making the matching, the inference engine would use the ontology, the digital rights, the policy and the trust expressed in OWL.

4 Conclusion and Future Work

We have proposed agent-based framework for virtual organizations in grid and introduced an ontology-based matchmaking mechanism in the framework. The agent-based approach can meet the requirements of scalable, dynamic, autonomous architecture of virtual organization. Ontology-based method can realize the semantic matching and fits in with the development of Semantic Web and Semantic Grid.

We just propose a fundamental to agent-based modeling for virtual organizations. There are a lot of future works to be done, such as, interoperation and integration of multi-level ontologies in a complex virtual organization, creation of the domain-specific ontology, the implementation of the model and the framework.

Acknowledgement

This paper is jointly supported by NSFC with project no. 60173036 and JSNSF with project no. BK2003001. The authors are also grateful to Huiying Li and Xiang Zhang for their suggestions for this work.

References

1. Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15 (3). 200-222. 2001. <http://www.globus.org/research/papers/anatomy.pdf>.

2. G. Wasson and M. Humphrey. Policy and Enforcement in Virtual Organizations. In 4th International Workshop on Grid Computing (Grid2003). (associated with Supercomputing 2003). Phoenix, AZ. Nov 17, 2003.
3. G. Wasson and M. Humphrey. Towards Explicit Policy Management in Virtual Organizations. In IEEE 4th International Workshop on Policies for Distributed Systems and Networks. Lake Como, Italy. June 4-6, 2003.
4. David W. Walker. The Grid, Virtual Organizations, and Problem-Solving Environments. In IEEE International Conference on Cluster Computing, California, USA, Oct. 8-11, 2001, <http://www.cacr.caltech.edu/cluster2001/program/talks/walker.pdf>
5. Tieyan Li, HuaFei Zhu. A Novel Two-level Trust Model for Grid. ICICS'03. Oct. 10-13, 2003. Springer LNCS 2836.
6. Farag Azzedin, Muthucumaru Maheswaran: Towards Trust-Aware Resource Management in Grid Computing Systems. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02), Berlin, Germany, May 21 - 24, 2002: 452-457.
7. M. J. Wooldridge. Intelligent Agents. In: G. Weiss, ed. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press. 1999, pp. 27-77.
8. F. Manola and C. Thompson. Characterizing the agent grid.
<http://www.objs.com/agility/techreports/990623-characterizing-the-agentgrid.html>,
9. O. F. Rana and D. W. Walker. 'The Agent Grid': Agent-based resource integration in PSEs. In Proceedings of the 16th IMACS World Congress on Scientific Computing, Applied Mathematics and Simulation, Lausanne, witzerland, August 2000.
10. T. R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199-220, 1993.
11. Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
12. Ian Horrocks. A proposal for an OWL Rules Language.
<http://www.cs.man.ac.uk/~horrocks/DAML/Rules/WD-OWL-rules-20031016/>
13. Youyong Zou. OWL content language for FIPA acl message. <http://taga.umbc.edu/>
14. M. Solomon R. Raman, M. Livny. Matchmaking distributed resource management for high throughput computing. In Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, IL, July 1998.
15. The portable batch system. <http://pbs.mrj.com>.

A Pattern-Based Approach to Facilitating Service Composition*

Haitao Hu^{1,2}, Yanbo Han^{1,2}, Kui Huang^{1,2}, Gang Li^{1,2}, and Zhuofeng Zhao^{1,2}

¹ Software Division, Institute of Computing Technology, Chinese Academy of Sciences
100080, Beijing, China

² Graduate School of Chinese Academy of Sciences, 100039, Beijing, China
{huhaitao, yhan, huangkui, ligang, zhaozf}@software.ict.ac.cn

Abstract. This paper presents a pattern-based approach to facilitating the composition of Web services, which enables business users to use composite services more effectively. With the support of patterns, business users can construct applications with larger-granularity components, amend and customize their own patterns to meet personalized requirements. The approach is illustrated with a case study. We suggest the patterns be used during the orchestration stage in a service composition process. By doing so, the composition logic built into the pattern can be made available to other users.

1 Introduction

Based on a stack of standards like WSDL, UDDI and SOAP [1], Web services provides an effective means for building up distributed applications, allowing us to integrate inter-organizational services in a loosely-coupled manner. Web services composition specifies constraints on how the operations of a collection of Web services and their joint behavior to fulfill more complex functionality. Some languages such as WSFL, XLANG, BPEL4WS and DAML-S [2-5] have emerged. The goal of them is to glue services together in a process-oriented way, using the basic constructs of sequence, splits, joins and iteration. It is still difficult for business users to comprehend and use them directly. What they expect is the ‘well-defined’ composition language which is easy to understand and use.

Meanwhile, it costs users much time to construct a service-oriented application, when they need to build the analogous application they have to orchestrate it again. It brings much trouble to users. In the dynamic and autonomous service environment, it is difficult to change from this orchestration mode into the objective, methodical and with tools supporting efficient composition modes.

Pattern-based approach may solve these problems well. Pattern not only can help users to design good, tested and extensible processes but also can be reusable. Users can compose the services without the tedious operation by selecting the corresponding predefined pattern and appending the simple operation to finish it. Moreover,

* This paper is supported by the National Natural Science Foundation of China under Grant No. 60173018 and the Young Scientist Fund of ICT, CAS under Grant No. 20026180-22.

pattern is independent of scale, videlicet, users can use the pattern with larger-granularity and different abstract hierarchy. Pattern provides a set of thinking in software development.

Aiming at facilitating users' usage, we present a pattern-based approach to making them compose services more effectively. The paper organized as follows. In section 2 some related works are analyzed; section 3 proposes several patterns to make composing web services easier and illustrates them in detail; section 4 shows the usage of them through a scenario; section 5 concludes and lists some future directions.

2 Related Works

Alexander describes a pattern as a three-part rule, which expresses a relation between a certain context, a problem and a solution [7]. Patterns can help to create a shared language for communicating insight and experience about the problems and their solutions in a particular context [8]. Since pattern is an idea that proved to be useful in one practical context and will probably be useful in others.

Combining workflow technology and design pattern, Meszaros and Brown present a pattern language for workflow systems [9]. They consider the workflow facility a component of a system and describe the process for creating the system. Van der Aalst et al. summarize workflow patterns and advanced workflow patterns in [10] and [11]. Based on these patterns they evaluate 15 workflow products and detect considerable differences in expressive power.

Moe Thandar Tut would like to propose the use of patterns during the planning stage of service composition [12]. Patterns represent a proven way of doing something. They could be business patterns such as how to model online store-fronts, or generic patterns such as project work patterns. His assumption is that the business goal is to successfully compose services, not to decompose the process model to the lower level.

The writers summarize some of the challenges and recent developments in the area of Web services integration in [13]. They abstract them in the form of software design patterns. Accordingly, they identify a collection of prospective patterns addressing various activities in the life cycle of a composite Web services. In [14], they provide an overview of several proto-patterns for architecting and managing composite web services. These proto-patterns aggregate results from previous efforts in the area of Web services composition, into guidelines for addressing design issues related to the various activities in the life cycle of a composite service. The contribution is a starting point towards a pattern-oriented service composition methodology.

3 Some Patterns for Service Composition

Patterns can be used to represent reusable business process logic. Every element in the pattern might be a service or another pattern. In the Fig. 1, we illustrate the difference between the traditional approach of service composition and the pattern-based approach. In the left part is the traditional approach. A, B and C are business services,

the users have to orchestrate them from-scratch every time. The right part is a pattern-based approach. D is a service and P is a pattern, which encapsulates the composition logic between the services and can be available to other users.

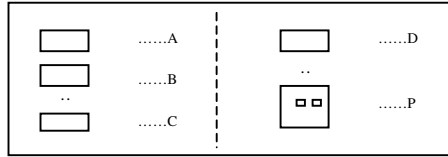


Fig. 1. Illustration of Web services composition pattern

In order to make full use of composite services more efficiently, we present two kinds of patterns for the different users: generic patterns and special patterns. Generic patterns can be used by users who know a little computer programming knowledge, we entitle them *power user*; special patterns can be used by users who don't know the knowledge within their specific domain, we entitle them *end user*.

3.1 Generic Patterns

We abstract five generic patterns by summarizing the most workflow products and the current prevalent Web services composition languages [2-6]. The five patterns lack orthogonality in theory but the most cases of services composition can be depicted by using one or more of them. Aalst has testified it. Moreover, they can be manipulated easily by users. The five patterns are sequence pattern, semi-sequence pattern, simultaneity pattern, exclusion pattern and repeat pattern. They are directly supported in the Flame2008 [15], we will explain it detailedly in section 4. In the following we suppose a and b stand for business service or a pattern.

Sequence pattern ($a \rightarrow b$): this is the simplest form of service composition. b is carried out after the completion of a in the same process. For example, the service *BookFlight* is executed after the completion of the service *QueryFlight*.

Semi-sequence pattern ($a \Rightarrow b$): a and b can be carried out at the same time in a process, but b must finish after a has finished. For example, the service *QueryWeather* and *BookTicket* can be executed at the same time but *BookTicket* must finish after *QueryWeather* had finished.

Simultaneity pattern ($a \leftrightarrow b$): a and b can be executed in parallel, thus allowing services to be executed simultaneously or in any order. For example, the service *RentCar* and *ReserveHotel* can be executed at the same time.

Exclusion pattern ($a \updownarrow b$): only one of the two services can be executed in the process, viz. a service in the process where, based on a decision or control logic, one of several branches is chosen. For instance, the service *ReserveSightseeing* and *BookTicket* only one can execute according to the service *QueryWeather*'s result.

Repeat pattern (a^n): a can be executed many times in the same process at some conditions. Such as, *QueryWeather* will be executed everyday in the whole journey.

The purpose of defining these patterns is to facilitate the business user's operation and can correspond with BPEL4WS [4], which has become the de-facto standard for Web services composition. Some structured activities (switch, sequence, etc.) were defined in BPEL4WS. We can change the application constructed by the patterns into BPEL4WS application on Flame2008 Platform easily.

3.2 Special Pattern

Besides the above generic patterns, we will present a special pattern for the *end user*: Time-Driven Pattern.

Services have some relations with control flow and data flow in a process, and the composite service is represented by process logic (switch, sequence, etc). There is a situation that the logic is weak and most services are triggered by time. For this kind of composite process we recommend the Time-Driven Pattern to construct it. The users only set some attributes of the time and drag their needed services from *service community*, which is a view of services, and then place them to proper location in the time dimension. It is so simple that can be understand and use by the *end user*.

This pattern defines *TimePeriod*, *TimeSlice*, *ServiceNeeded* to implement services composition. *TimePeriod* defines an interval that a business process goes through, which gives the start time of the process: *P_StartTime*, and the end time: *P_EndTime*. They can be defined with absolute time or relative time. *TimePeriod* comprises some *TimeSlices*, *TimeSlice* is time unit in the *TimePeriod*, Which includes start time and end time denoted with *S_StartTime* and *S_EndTime*. They are relative time compared with the *P_StartTime* or *P_EndTime*. *TimeSlice* is the basic time unit, every *TimeSlice* includes one or several services, which denoted by *ServiceNeeded*.

ServiceNeeded defines the services which used at a *TimeSlice*. *ServiceNeeded* includes an attribute: *Precondition*, which defines the precondition of the service which is used. If the precondition is null, it means the service will be executed when the time arrives. If there are several services in a *TimeSlices* the users can set the *Precondition* to realize which service should be executed. Moreover, *ServiceNeeded* also include *Suppress* and *Continue* attributes, a service can span a *TimeSlices* by setting the two attributes. Each value of the attributes can be 'yes' or 'no', default is 'no'. If *Suppress's* value is 'yes', it denotes the end of the *TimeSlice* is independent of the service's end. If *Suppress's* value is 'no', it denotes the end of the service will influence the *TimeSlice's* end. The *Continue's* value is 'yes', it denotes the service is inherited the preceding *TimeSlice*, and the *Continue's* value is 'no', it denotes the service begins in a new *TimeSlice*.

3.3 Translating the Application from Special Pattern to Generic Patterns

In this subsection we will illustrate how to translate an application from special pattern to generic patterns. There are no fundamental difficulties but the reverse is more problematic. We will explain it in the following four cases:

There is only a service in a *TimeSlice* and the *Precondition* is null. The services between the different *TimeSlices* will be executed with sequence pattern.

There is only a service in a *TimeSlice* and the *Precondition* isn't null. Generally speaking the services between the different *TimeSlices* will be executed in sequence, but not every service can be enabled in the process, it's restricted by the *Precondition*.

There are several services in a *TimeSlice* and the *Precondition* is null. In this case the services between the different *TimeSlices* still be executed with sequence pattern, and the services in the same *TimeSlice* will be executed with simultaneity pattern.

There are several services in a *TimeSlice* but the *Precondition* isn't null. In this case, the services between the different *TimeSlices* still be executed with sequence pattern, and the services in the same *TimeSlice* will be executed with simultaneity pattern or exclusion pattern, which is decided by the *Precondition*.

Here we don't consider how to translate the special pattern into repeat pattern. In fact the user can place the same service in some continuous *TimeSlices* to denote the service be executed repeatedly in the special pattern, but it is troubled to the user. How to extend the expressive power of the *Precondition* will be our future work.

4 Implementation in FLAME2008

In this section, we illustrate how to orchestrate the services by using the pattern-based approach on the Flame2008 Platform.

4.1 Translating the Application from Special Pattern to Generic Patterns

Flame2008 is a platform, which is abbreviated from *A Flexible Semantic Web Service Management Environment for the Olympic Games Beijing 2008* [15][16]. On the platform an effective information system providing personalized and one-stop information services to the general public should be based. The front-end of the platform is business-level programming environment. Adopting the service-oriented paradigm, we design the service integration with both generic patterns and special pattern to mediating between diverse, rapidly changing user requirements and composites of individual services scattered over the Internet. In the next subsection we give an example constructed by the pattern-based approach on the platform.

4.2 A Usage Scenario

Mr. George, who is an American news agency reporter, staying in Hongkong. On August 14th, 2008, he receives the notice to interview the American baseball players on August 18th, 2008. He wants to use the Flame2008 Platform to arrange his whole journey. According to George's requirement, the following services may be included:

- Reserving the return ticket leaving for Beijing from Hongkong on 2008-08-15 and return Hongkong on 2008-08-20.
- Reservations for the hotel in Beijing from 2008-08-15 to 2008-08-19.
- Reserving the service of interview on 2008-08-18.
- In order to acquire the weather status in time, he reserves the Forecast Service.
- George will finish his interview work on 2008-08-18. He decides to visit the Summer Palace if the day is sunny, otherwise, he will watch the football match between his favorite Brazil Team and American Team.

Fig. 2 is the illustration of his whole journey, a0 and a7 denotes the start and end; a1, a2, a3, a4, a5 and a6 are the services: *OrderAirplaneTicket*, *OrderAccommodation*, *ArrangeInterview*, *InquiryWeather*, *OrderMatchTicket* and *OrderSightseeingTicket*.

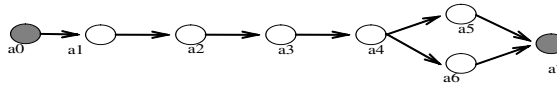


Fig. 2. Flow chart of George's journey

George searches the repository and finds a pattern: $a1 \rightarrow a2 \rightarrow a5$. It is a summarization of many travel cases. There are two patterns in it. One is p1 ($a1 \rightarrow a2$) and the other is p2 ($p1 \rightarrow a5$). Obviously, the pattern can't meet George's personalized requirement completely. So he decides to amend it. He amends p1 to be ($a1 \rightarrow a2 \rightarrow a3$) and p2 to be ($p1 \rightarrow a4 \Rightarrow (a5 \uparrow a6)$), thus the amended pattern ($a1 \rightarrow a2 \rightarrow a3 \rightarrow a4 \Rightarrow (a5 \uparrow a6)$) can meet his requirement totally. Fig. 3 is the graphical representation of this pattern in Flame2008.

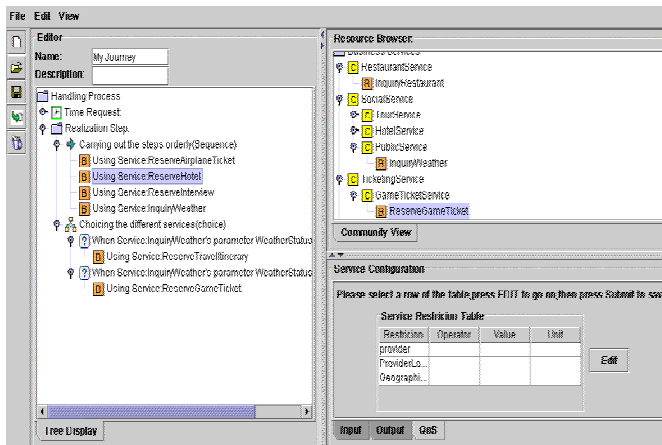


Fig. 3. Using Generic Pattern to compose services

Fig. 4 shows the example of using Time-Driven Pattern to construct the application; George sets the start and end of the *TimePeriod* and the *TimeSlice*, drags the services and places the right *TimeSlice*. The logic relation between the services expressed by setting the services attributes. In the version1.0 of the platform we have not implemented the graphical representation of the special pattern, it is our future work.

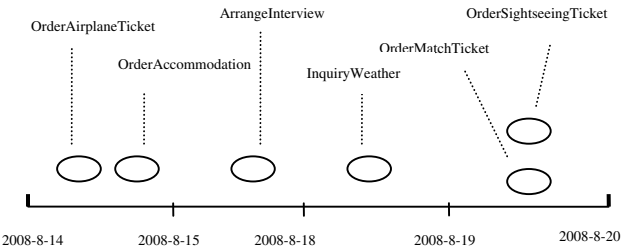


Fig. 4. Example of using Time-Driven Pattern

Table 1 is the XML fragment of using Time-Driven Pattern, and the Mediation of the platform can parse it to BPEL4WS file.

Table 1. The XML fragment of using Time-Driven Pattern

```
<? xml version="1.0" encoding="UTF-8"?>
<Process>
  <BizServices>
    <BizService name="OrderAirplaneTicket">
      .....
    </BizService>
  </BizServices>
  <TimePeriod p_startTime="2008-08-15" p_endTime="2008-08-20">
    <TimeSlice name="slice1" s_startTime="2008-08-15">
      <ServiceNeeded ref="OrderAirplaneTicket" suppress="no" continue="no">
        .....
      </ServiceNeeded>
    </TimeSlice>
    .....
    <TimeSlice name="slice4" s_startTime="afterSlice3">
      <ServiceNeeded ref="OrderMatchTicket" suppress="no" continue="no">
        <Precondition> ..... </Precondition>
      </ServiceNeeded>
      .....
    </TimeSlice>
  </TimePeriod>
</Process>
```

5 Conclusions and Future Work

Web services composition is a complex field, and there is no simple ‘cookbook’ answer. That’s why pattern is a useful way to convey experience that usually only lives in designers’ heads. In this paper we have raised the question of how to compose services with patterns in the orchestration phase. We have attempted to describe the generic patterns and special pattern could be used with an individual journey example.

In order to realize Web services composition more effectively, we should take into account other related patterns. Such as user requirement presentation pattern, services selection pattern, and the interaction negotiation pattern when the users’ requirement could not be met, etc. These patterns can meet the users’ requirement from the different aspects. As a step further in this direction, our ongoing work aims at providing a web services composition pattern language or a pattern family.

References

1. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S.: Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6(2):86–93, March 2002.
2. Web service flow language (WSFL) 1.0, : <http://www-4.ibm.com/software/solution/Webservices/pdf/WSFL.pdf>, accessed 22 September 2002.
3. Web Services for Business Process Design,: http://www.gotdotnet.com/team/xml_wsspecs/clang-c/default.htm, accessed 22 September 2002.
4. BPEL4WS. Business Process Execution Language for Web Services, Version 1.1. <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>, 2003.
5. Houstis, C., Lalis, S., Pitikakis, M., Vasilakis, G., Kritikos, K., Smardas, A.: A grid service-based infrastructure for accessing scientific collections: The case of the ARION system, *The International Journal of High Performance Computing Applications* special issue on Grid Computing: Infrastructure and Applications, to appear August 2003.
6. Van der Aalst, M. Dumas, AHM ter Hofstede. Web Service Composition Languages: Old Wine in New Bottles? *Proceeding of the 29th EUROMICRO Conference: New Waves in System Architecture*, pages 298-305. IEEE Computer Society, Los Alamitos, CA, 2003.
7. Alexander, C.: *The Timeless way of Building*, Oxford University Press, New York. Appleton, 1979.
8. B. Appleton, *Patterns and Software: Essential Concepts and Terminology*. <http://www.enteract.com/~bradapp/docs/patterns-intro.html#PatternElements> accessed on 7 Mar 2002.
9. G. Meszaros and K. Brown, A Pattern Language for Workflow Systems. *Proceedings of the 4th Pattern Languages of Programming Conference*, Washington University Technical Report 97-34 (WUCS-97-34), 1997
10. Van der Aalst, Barros, Hofstede, *Advanced Workflow Patterns*. Source, *Lecture Notes In Computer Science archive Proceedings of the 7th International Conference on Cooperative Information Systems* table of contents. Pages: 18 – 29, 2000.

11. Van der Aalst, A.H.M ter Hofstede, et al, Workflow Patterns.QUT Technical report, FIT-TR-2002-02. Queensland University of Technology, Brisbane, 2002.
12. Moe Thandar Tut and David Edmond, The Use of Patterns in Service Composition. <http://www.fit.qut.edu.au/~edmond/papers/wsebt02.pdf>
13. B. Benatallah, M. Dumas, MC Fauvet, FA Rabhi, Towards patterns of web services composition. In Patterns and Skeletons for Parallel and Distributed Programming, F. Rabhi and S. Gorlatch, Eds. Springer Verlag, London, UK. 2002.
14. B. Benatallah, M. Dumas, et al, Overview of Some Patterns for Architecting and Managing Composite Web Services, ACM SIGecom Exchanges, Volume 3.3, pp. 9-16, 2002.
15. B. Holtkamp, Y. Han, et al, FLAME2008-Personalized Web Services for the Olympic Games 2008 in Beijing, Proceedings of Exchallenges 2003, Bologna, Italy, Oct. 2003.
16. Y. Han, et al, CAFISE: An Approach Enabling On-Demand Configuration of Service Grid Applications, Journal of Computer Science and Technology, Vol.18, No.4, 2003.

Automatic Service Matching and Service Discovery Based on Ontology

Chuan Lin^{1,2}, Zhaohui Wu², Shuiguang Deng², and Li Kuang²

¹ Wenzhou Normal College, Wenzhou 325000 PRC
linflag@163.com

² College of Computer Science, Zhejiang University, Hangzhou 310027 PRC
{wzh,dengsg}@zju.edu.cn, kuangli2000@163.com

Abstract. With the development of standards and technologies related to services, more and more services are becoming available on the Internet. To find a target service from tremendous ones, which provides the wanted functions, has become more and more difficult. Thus automatic service matching and service discovery become important. In order to realize the automatic service matching and service discovery, ontology is the key solution. In this paper, we use ontology to describe services and use ontology to compute the similarity between two concepts. Based on the computation of ontology, we propose algorithms to search target services based on ontology information embedded in them. Our algorithm cares about not only the relationship between ontology classes, but also the relationship between classes and properties. Thus it provides more accuracy than other related methods.

1 Introduction

Web service has become a most suitable solution for e-business application now [1]. Because of web service self-contained and self-described, it can be published, discovered and invoked. When a function that cannot be realized by the existing services is required, the existing services can combined together to fulfill the request. The dynamic composition of services requires the location of services based on their capabilities and the recognition of those services that can be matched together to create a composition. Web service technology is supported by UDDI and WSDL, but UDDI and WSDL are less of semantic information. The semantics of Web services is crucial to enabling automatic service composition. It is important to insure that selected services for composition offer the “right” features. To help capturing Web services’ semantic features, we use the concept of ontology [2]. Ontology provide potential terms for describing our knowledge about the domain[3]. They are expected to play a central role in the Semantic Web, extending syntactic service interoperability to semantic interoperability[4].

As a part of the DART-GRID [5] project in Zhejiang university, we are using technologies from semantic web, web service, and our earlier research in workflow management [6] to build a framework “DartFlow” composing services dynamically and executing automatically. The remainder of this paper is organized as follows. We present an outline of a framework- DartFlow . Then, we discuss a matching algorithm in DartFlow to search target services based on ontology information. Later on, some related work is presented. The last is our conclusion and future work.

2 Overview of DartFlow

We use workflow technology in DartFlow to compose services dynamically. We call a service composition “Serviceflow”. Key Features of DartFlow[7] are as follows.

- Enriches Service and service composition semantically.
- Realizes efficient and flexible composition of service.
- Allows dynamically change partners and services by later binding of services.
- Ensure the successful execution through handling exceptions while invoking.

The architecture of DartFlow is illustrated in fig.1.

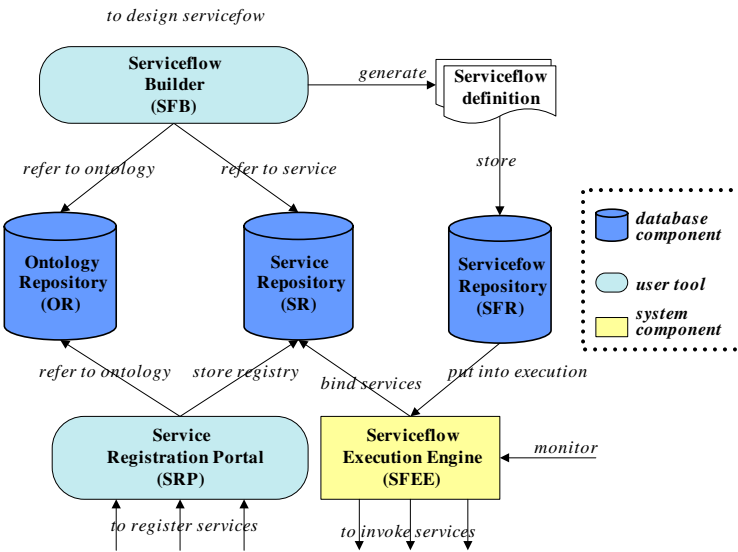


Fig. 1. The Architecture of DartFlow

3 Semantic Description and Similarity of Web Services

Most of existing web services are described in WSDL in Internet. But there is a lack of semantic knowledge in WSDL. If adding semantic knowledge into each of operations and messages described in abstract service interface, the services can understand each other. In DartFlow, we use ontology to enforce the semantics. Every message, operation described in WSDL must be associated with a specific class already defined in service ontology. These associated information and some other necessary information, such as service access methods, are saved in an OSDL (ontology service description language) file. This information is used to register, discover, match and invoke a service automatically. Figure 2 shows an example of OSDL file.

According to the existed abstract services flow, the services composition requires the location of concrete services based on their capabilities and the recognition of those abstract services that can be matched. There are mainly two services matches:

```

<OSDL OntologySpace="http://ccnt.zju.edu.cn/Ontology/TouristOntology">
  <portType Name="WeatherReport">
    <Operation Name="Weather" OntologyClass="WeatherReport">
      <input>
        <part name="Location" OntologyClass="City"/>
        <part name="Day" OntologyClass="Date"/>
      </input>
      <output>
        <part name="WeatherReturn" OntologyClass="Centigrate"/>
      </output>
    </Operation>
  </portType>
  <URL>http://localhost:8080/axis/Weather.wsdl</URL>
</OSDL>

```

Fig. 2. An Example of OSDL File

match for services functions and match for fundamental messages. By using of the semantic description of the services function in the OSDL file, programs can discover the services based on comprehension. Supposing the set of input messages and output messages of the abstract service to be inR and $outR$, the set of input messages and output messages of the concrete service to be inA and $outA$, we consider the services will match while $outA \supseteq outR$ and meanwhile, $inR \supseteq inA$, that is to say, $outA$ can supply all the outputs that $outR$ needs and at the same time inR can satisfy all the inputs that inA needs.

The match above can be considered as a question to judge whether the set A contain the set B , that is to say, for every element- b in the set B , to judge if there is an element- a in the set A which semantically similar to element- b . It needs to judge with the aid of the concept of ontology.

3.1 Relevant Definitions

Definition 1. Semantic graph $G(V, E)$. G is a 2-tuple, V is the set of finite and non-empty vertexes, and each vertex represents a class in service ontology or a data type. E is the set of the relations between the two vertexes. While vertex Y is the subclass of the vertex X , there is a directed real edge from vertex X to vertex Y . While vertex Y is the object property of the vertex X , there is a directed dashed edge from vertex X to vertex Y , there is no dashed connection between the subclass of vertex X and the vertex Y . While vertex Y is the data type property of vertex X , there is a direct dash-dotted edge from vertex X to vertex Y . Figure 3 is an example of Semantic Graph.

Definition 2. Property Degree of Vertex $PropertyNum(a)$. In the graph $G(V, E)$, it is the amount of the direct dashed edges and direct dashdotted edges from the vertex a .

Definition 3. Inheriting Vertex Set $Vexl(a, b)$. In the graph $G(V, E)$, if there exists the path made up of the direct real edges from vertex a to vertex b , then set $Vexl(a, b)$ should consist of all the vertexes in the path.

Definition 4. Property Vertex Set $Vex2(a, b)$. In the graph $G(V, E)$, if there exists the path made up of the dashed edges and dashdotted edges from vertex a to vertex b , then set $Vex2(a, b)$ should consist of all the vertexes in the path but the vertex b .

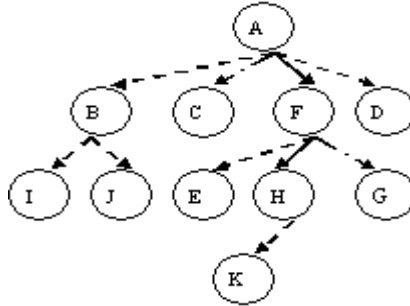


Fig. 3. An Example of Semantic Graph

3.2 Semantic Comparison

From Definition 1, we can give semantic comparison between Class X and Class Y once they have one of following four relations:

1. Same class;
2. Inherited relation: Class X is the subclass of Class Y, e.g. in the Fig.3, vertex F is the subclass of Class A, vertex H is the subclass of Class A.
3. Property relation: Class Y is the property of Class X, e.g. in the Fig.3, vertex B is the property of Class A, vertex I is the property of Class A.
4. Mixed relation: Class X is the subclass of Class Z, Class Y is the property of Class Z, the relation of Class X and Class Y is called Mixed relation e.g. in the Fig.3, vertex F and B.

3.3 Semantic Similarity Computing

In service ontology, a class is only represented by its properties, the value of function $Similarity(X, Y)$ is the match degree of Class X to Class Y, when Class X can offer all the properties that Class Y can offer, we call the Class X match to Class Y, when Class X can only offer part properties of what Class Y can offer, we call the two classes is part matched. The value of $Similarity(X, Y)$ ranges at $[0, 1]$. While “0” means there are no similarity between X and Y at all; but “1” means they are the same indeed. $Similarity(X, Y)$ and $Similarity(Y, X)$ represent different match.

The similarity between two classes can be computed from the semantic graph:

1. Same class: The match degree of Class X with Class Y is $Similarity(X, Y) = 1$ and $Similarity(Y, X) = 1$.
2. Inherited relation: there must have a real edge between two vertex. Suppose Class Y is the subclass of Class X, Class X is the subclass of Class Z if existed, then Class Y can supply what the Class X needs, Class X can give part of what Class Y needs.

- The match degree of Class Y to Class X is $Similarity(Y, X) = 1$, e.g., in the Fig.3, $Similarity(H, A) = 1$.
- The match degree of Class X to Class Y is

$$Similarity(X, Y) = \frac{\sum_{node \in Vex1(X, Z)} PropertyNum(node)}{\sum_{node \in Vex1(Y, Z)} PropertyNum(node)},$$

e.g., in the Fig.3, $\sum_{node \in Vex1(H, A)} PropertyNum(node) = 3 + 2 + 1 = 6$,

$\sum_{node \in Vex1(A, A)} PropertyNum(node) = 3$, $Similarity(A, H) = 3/6$.

3. Property relation: there must have a dashed edge between two vertex. If Class Y is the property of Class X, then Class X can supply what Class Y needs, Class Y can give part of what Class X needs.
- The match degree of Class X to Class Y is $Similarity(X, Y) = 1$, e.g., in the Fig.3, $Similarity(A, I) = 1$.
- The match degree of Class Y to Class X is

$$Similarity(Y, X) = \frac{1}{\prod_{node \in Vex2(X, Y)} PropertyNum(node)},$$

e.g., in the Fig.3, $Similarity(I, A) = 1/(2 \times 3) = 1/6$.

4. Mixed relation: Class X is the property of Class Z and Class Y is the subclass of Class Z.
- The match degree of Class Y to Class X is $Similarity(Y, X) = 1$, e.g., in the Fig.3, $Similarity(H, B) = 1$.
- The match degree of Class X to Class Y is

$$Similarity(X, Y) = \frac{1}{\prod_{node \in Vex2(Z, X)} PropertyNum(node)} \div \frac{\sum_{node \in Vex1(Z, Y)} PropertyNum(node)}{\sum_{node \in Vex1(Z, Y)} PropertyNum(node)},$$

e.g., in the Fig.3, $\prod_{node \in Vex2(A, B)} PropertyNum(node) = 3$,

$\sum_{node \in Vex1(A, H)} PropertyNum(node) = 6$, $Similarity(B, H) = 1/18$.

5. When relations above don't exist, the match degree of Class Y with Class X is "0".

3.4 Algorithm to Judge Whether the Set A Contains Set B

For each element-b of the set B, in the set A finding out element-a whose similarity value to b is the maximum and the value is not 0, then record the value. Sum up all the maximum similarities value of the set A to each elements in the set B and get the

average, the value shows the degree of the set A contains the set B. If exists element-b in the set B, the similarity of any element-a in the set A to b is 0, then the set A does not contain the set B. The algorithm is shown in fig.4.

```

/*to judge whether the set A contains the set B */
subsume(A,B){
    #define NO-MATCH 0
    int n=1;           //the serial number of the elements in the set B
    double sim=NO-MATCH; //the similarity of the set A to B
    double maxsimilarity[]={NO-MATCH,...}; //record the maximum
        //similarity of elements in the set B to elements in the set A
    for (each of element-b in the set B)
    {
        for (each of element-a in the set A)
        {
            if (similarity(a,b) > maxsimilarity[n])
                maxsimilarity[n]=similarity(a,b);
            if (maxsimilarity[n]==0)
                {sim= NO-MATCH; break;}
            else {n++; sim+=maxsimilarity[n];}
        }
    }
    if (sim<= NO-MATCH) sim=sim/n;
    return sim;
}

```

Fig. 4. An algorithm to Judge Whether the Set A Contains Set B

The web service of an advertisement and the service request can be fuzzy-matched by using algorithm above. The corresponding semantic similarity can be calculated.

4 Service Match Computing

During the service match, we should consider not only the match for service function, but also the match for basic service information and other properties, for example, Qos etc., of course the match for the function is the most important. An algorithm to compute the match between two services is illustrated in fig.5.

```

/* service match computing */
match(request){
    matchSet=empty list; //the set of matched service
    for (each of adv services in register repository) {
        if ( ( the amount of output messages of adv > the amount of output
            // messages needed by request )
            if (the amount of input messages of request > the amount of input
            // messages needed by adv)
                if (subsume(outA,outR) <= NO-MATCH)
                    if (subsume(inR,inA) <= NO-MATCH)
                        matchSet.insert(adv); } //insert the matched ads service
        // to matchSet according to the descending similarity
    }
}

```

Fig. 5. An algorithm to Compute the Match between Two Services

Assuming there are two services *adv1* and *adv2* matched with service *ads*, sort them according to the matches degree with *ads* service. The matched service can be inserted to *matchSet* with the descending similarities according to the following :

if ($\text{subsume}(\text{adv1.outA}, \text{outR}) > \text{subsume}(\text{adv2.outA}, \text{outR})$) *adv1* in front of *adv2*

if ($\text{subsume}(\text{adv1.outA}, \text{outR}) = \text{subsume}(\text{adv2.outA}, \text{outR})$ &

$\text{subsume}(\text{inR}, \text{adv1.inA}) > \text{subsume}(\text{inR}, \text{adv2.inA})$) *adv1* in front of *adv2*

if ($\text{subsume}(\text{adv1.outA}, \text{outR}) = \text{subsume}(\text{adv2.outA}, \text{outR})$ &

$\text{subsume}(\text{inR}, \text{adv1.inA}) = \text{subsume}(\text{inR}, \text{adv2.inA})$) *adv1* and *adv2* juxtapose

5 Related Work

Among the present service discovery approaches, most are frame-based [8, 9, 10], e.g. UDDI. All the commercial service search technologies we are aware of (e.g. Jini, eSpeak, Salutation, UDDI) use the frame-based approach [9, 10], typically with an at least partially pre-enumerated vocabulary of service types and properties. The frame-based approach is taken one step further in the deductive retrieval approach [11] wherein service properties are expressed formally using logic. This approach, however, faces two very serious practical difficulties. One difficulty is to model the semantics of non-trivial queries and services using formal logic, the other is that the proof process implicit in this kind of search, which has a high computational complexity, makes it extremely slow. To make up for this, [8] proposes an approach for service discovery on the semantic web by using process ontologies. But process ontology only describes semantically based process and has not domain knowledge, so there is a limit in related service matching. In addition, this approach requires the web service description with special mode instead of standard WSDL. The technique of semantic web service discovery based on ontology [12] puts forward the ontology-based matching algorithm of web service, which uses DAML-S [13] for service description. Matching algorithm only pays attention to ontology inherits relation, having no consideration for matching between properties and classes. Base on domains ontology, [14] gives an approach for semantic discovery and matching of Web services. It is hard to organize UDDI in a P2P network since UDDI is complicated.

6 Conclusion and Future Work

Adding domain semantic information to WSDL based on ontology can realize the intelligent search of service. Sufficient semantic information can make search more efficient and more accurate. Our matching algorithm provides a way for automatic dynamic discovery, selection and matching services. Besides the semantic information mentioned in this paper, we would give some further researches such as QOS in service discovery and service matching.

Acknowledgement

This work is supported in part by China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and

service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid : Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

References

1. Feisi Science and Technique Center of Research and Development. Application and Development of Java Web Service. Beijing Publishing House of Electronics Industry, 2002. 190-196
2. Brahim Medjahed, Athman Bouguettaya, Ahmed K. Elmagarmid. Composing Web services on the Semantic Web. The VLDB Journal (2003)
3. Chandra solaran B, Josephson J R. What Are Ontologies, and Why Do we Need Them? IEEE Intelligent systems, January/February 1999
4. Horrocks I (2002) DAML+OIL: a description logic for the Semantic Web. IEEE Data Eng Bull 25(1):4-9
5. Z.H. Wu, H.J. Chen. DartGrid: Semantic-based Database Grid. Accepted to International Conference on Computational Science. 2004.
6. S.G. Deng, Z. Yu, Z.H. Wu. Enhancement of Workflow Flexibility by Composing Activities at Runtime. Proc. Of the Annual ACM Symposium on Applied Computing. 2004
7. Shuiguan Deng, Zhaohui Wu, Chuan Li. A Framework for Managing Serviceflow in a Dynamic Flexible Way. Submitted to WISE 2004
8. Klein M, Bernstein A. Searching for services on the semantic Web using process ontologies. In: Isabel C, ed. Proc. of the Int'l Semantic Web Working Symp. (SWWS2001). Amsterdam: IOS Press, 159-172.
9. Richard, G.G., Service advertisement and discovery: enabling universal device cooperation. IEEE Internet Computing, 2000. 4(5): p. 18-26.
10. ESPEAK, Hewlett Packard's Service Framework Specification. 2000, HP Inc.
11. Kuokka, D.R. and L.T. Harada, Issues and extensions for information matchmaking protocols. International Journal of Cooperative Information Systems, 1996. 5: p. 2-3.
12. Massimo Paolucci, Lakahiro Kawamura, Terry R. Payne, Katia Sycara. Semantic Matching of Web Services Capabilities
13. DAML Joint Committee. Daml+oil (march 2001) language. <http://www.daml.org/2001/03/daml+oil-index.html>, 2001
14. METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services

An Algorithm for Calculating Process Similarity to Cluster Open-Source Process Designs*

Kui Huang, Zhaotao Zhou, Yanbo Han, Gang Li, and Jing Wang

Institute of Computing Technology, Chinese Academy of Sciences
Graduate School of the Chinese Academy of Sciences
100080, Beijing, China
{huangkui, zzt, yhan, gangli, wangjing}@software.ict.ac.cn

Abstract. This paper proposes an algorithm for calculating process similarity in order to cluster process designs. A weighted graph is introduced for comparing processes in the intermediate form. The graph similarity is the weighted sum of similarity between sets of services and sets of service links that can be calculated based on the service similarity. The evaluation and application of the algorithm is discussed at the end of this paper.

1 Introduction

Web services are generally considered suitable for dynamic B2B (business-to-business) interactions with services deployed on behalf of other enterprises or business entities. They are now also widely used for B2C (business-to-consumer) applications [1] to meet the customer's personalized demands, especially in form of composition of Web services. In some commercial areas of B2C, such as travel service, it is not necessary to hide the inner process of a service such as a journey process. For example, various scheduling services for sightseeing can be registered as Web services, and travel agencies can combine them and design personalized travel processes, based on the experience of enterprise's history, to meet the traveller's demand. In this situation, an enterprise provides some processes, and supports the modification of processes for customers, as in [2]. There are several services composition languages, such as BPML, BPEL4WS, to assemble Web services. We call the corresponding assemblies process designs.

When a process design from an enterprise is provisioned to customers, customers can modify it on their demands and get a new design, which may meet not only their demands but also others'. If the enterprise collects these processes created by customers, they can mine customers' real requirements [3], and also make recommendations for the customers to share open-source process designs with each other. However, process designs are large-scale because every customer can produce many processes

* The research work is supported by the National Natural Science Foundation of China under Grant No. 60173018, the Key Scientific and Technological Program for the Tenth Five-Year Plan of China under Grant No. 2001BA904B07.

by modifying each existed process design, therefore it's not easy to directly mine customers' requirements from large-scale processes and not suitable for enterprises to recommend all process designs to customers. So it is necessary to reduce the large amount of raw process designs by categorizing them into smaller sets of similar items, which means clustering.

There are two kinds of clustering methods in informational retrieval [4]. One is based on measurement of similarity between the objects and the other proceeds directly from the object descriptions. Because process designs are composed of Web services and have complex structures without unique comparable descriptions, the second kind of method isn't fit. The only way to cluster them is based on a measure of similarity between them. In order to reuse the well-known clustering methods, we must find out how to measure the similarity of different process designs. But it is obscure to grasp the real meaning of likeness or unlikeness between them. Our paper focuses on choosing an appropriate measure (in the measure-theoretic sense) like all other measures of association in information retrieval [4].

The approach [5] in research of workflow mining [3] refers to the clustering of process traces or logs based on k-means clustering. This method focuses on the execution trace of processes and isn't suitable for the static structure of process designs. There is another special method, called item-to-item collaborative filtering [6], to cluster products using customers' purchased and rated information. Such a method avoids comparison with the concrete contents of objects and is suitable for all objects. But it depends on the purchased data and can't run before many customers use the products. The author in [7] introduces a graph of linked concepts to represent and cluster objects, which is similar to process designs. We extend such an idea to introduce a weighted graph to abstract a process design in our paper. A weighted graph is composed of two sets of respectively services and service links. Based on the measures of functions of Web services [8] and fuzzy sets [9], we can finally figure out the graph similarity that approximately represents the process similarity.

The rest of this paper is organized as follows: the following section analyzes the similarity between process designs and proposes a weighted graph to simplify processes. Section 3 describes a new algorithm of calculating similarity coefficient of graphs. The evaluation and application of the algorithm are described in Section 4. Section 5 provides some concluding remarks.

2 Analysis of Process Similarity

This section will analyze the similarity between process designs. There are several researches about the similarity between graphs in informational retrieval. In [7] each graph, called a *query graph*, sets a word at each node; the association between words is represented as a link. The similarity between graphs can be based on the linear combination of the inner product of the term significance (node) vectors and that of the term-term association (link) vectors.

Ideas in [7] are also suitable for process design. A process design is usually modeled by a graph such as Petri Nets and State charts. Aalst in [10] captures five elementary aspects of process control: Sequence, AND-split, AND-join, XOR-split

and XOR-join, which can express all of the process controls of language BPEL4WS or BPML. In order to compare different processes according to the same form, process design will be translated into a special graph like query graph in [7], called a *weighted graph*, denoted by $\langle N, L \rangle_{WG}$, in which each node in set N represents a Web services, and each link with a weight in set L represents a partial relation between neighbor services. A link between Service A and B, denoted by $\langle A, B \rangle$, means that B starts to run next after A is finished. Each link has a weight w to represent the probability of starting B after finishing A. $w \in [0, 1]$.

From [11], we can regard a process design as a graph composed of Web services and controls that include five elements. If we can transform these five controls into links that represent partial relations between neighbor services, process design will be transformed into a weighted graph. Sequence can be directly transformed to a link with a weight "1". Unfortunately, the weight of other controls depends on some conditions of process design and can't be accurately specified beforehand. In order to automatically calculate the process similarity, we approximatively specify the average weight of links belonging to the same XOR-split or XOR-join. The detailed transformation is described in the following algorithm:

Table 1. Algorithm for abstracting weighted links from a process

Begin

For each Sequence, two neighbor services compose a link with Weight "1"

For each AND-split, AND-join, XOR-split and XOR-join, each split or join is regarded as a link with the initial weight "1"

Searching from beginning of Web process to ending

When XOR-split with N cases appears in Location L1, searching backwards for corresponding join

If join exists in Location L2, then get a set of links for each case from L1 to L2

If join doesn't exist, then get a set of links for each case from L1 to ending or to next join L3

For each link with weight w in each set, weight of the link is adjusted to w/N

End when

When XOR-join with N cases appears in Location L4, searching ahead for corresponding split

If split is AND-split in Location L5, then get a set of links for each case from L5 to L4

For each link with weight w in each set, weight of the link is adjusted to w/N

End when

End searching

End

In the above table, the links for all of two neighbor services are firstly evaluated as weight "1". Secondly the special links will be disposed by searching from the beginning to the end. i) For XOR-split, each split has a condition. In fact the probability of executing each path is different from others and not known before running. The weight of links in each case is simply averaged by the number of splits in order to automatically compare them before running. This simplification is accessible. ii) For each path from AND-split to XOR-join, which means that process waits for one of the incoming splits to complete in XOR-join before activating the subsequent activity of XOR-join [11]. Simplification here is same to i).

After transforming a Web process into a weighted graph, the left is to calculate the similarity between sets of nodes and sets of links.

3 Similarity Coefficient Between Graphs

A weighted graph is composed of a set of nodes and a set of links. The similarity between weighted graphs is based on the linear combination of the similarity between sets of nodes and that of sets of links. Firstly, Web service is the basic element of a graph and similarity between services is very important. The details will be described in Section 3.1. Secondly, The similarity between sets of services is described in Section 3.2 as Node-based similarity coefficient, and the similarity between sets of links is described in Section 3.3 as Link-based similarity coefficient. The whole algorithm will be described in Section 3.4.

Before presenting the detailed algorithm, we firstly define two graphs used in the following section. Supposing that graph $\langle N_p, L_p \rangle_{WG}$ represents a process design P, while $N_p = \{P_1, P_2, \dots, P_m\}$, and P_i is a Web service ($1 \leq i \leq m$), $L_p = \{Lp_1, Lp_2, \dots, Lp_n\}$, Lp_j ($1 \leq j \leq n$) is a link of two neighbor services among N_p and weight of Lp_j is denoted by w_j^p . A graph $\langle N_q, L_q \rangle_{WG}$ as process Q is similar to P.

3.1 Similarity Coefficient Between Web Services

There are several researches about the similarity coefficient between services. In VISPO [8], e-Services are classified according to similarity-based and behavior-based analysis for substitutability purposes. When computing the similarity coefficient of services, this paper comprehensively considers several criteria including the descriptors and the semantic information of services, operations that can be invoked, messages and data exchanged. The Global similarity coefficient of two services, P_i and P_j denoted by $GSim(P_i, P_j)$, is described in details in [8]. It is not our focus, but we will regard $GSim()$ as the basis of other similarity coefficients in our paper.

3.2 Node-Based Similarity Coefficient

The similarity between sets has been researched in fuzzy mathematics [12]. The paper [9] describes degree of similarity relationship between two imprecise data that is similar to our problem. Such a method is also suitable for two sets of services.

The membership function of a set S, denoted by $F_s()$, is the mapping from the discrete input to the degrees of membership between zero and one, which means the degree of this input belonging to the set. For example, if a service p exists in a set of services S, $F_s(p)$ will be evaluated by one. The membership function of a set P denoted by $\{P_1, P_2, \dots, P_m\}$ can be generally evaluated by the maximum values among the similarity coefficients between service p and every service of the set, denoted by $\text{Max}\{Gsim(p, P_i)\}$. Extending this idea, the membership function mapping from a set to the range of [0,1], which is called fuzzy conditional probability relation in [9],

can represent the degree of a set belonging to another set, which is calculated by averaging all the degrees of services of this set belonging to another set.

According to the method described in [9], the Node-based similarity coefficient of two sets of Web services N_p and N_q , denoted by *NodeSim* (N_p , N_q), is given as follows:

$$\text{NodeSim}(N_p, N_q) = \frac{\sum_{i=1}^m \text{Max}_{k=1}^l \{GSim(P_i, Q_k)\} + \sum_{j=1}^l \text{Max}_{k=1}^m \{GSim(P_k, Q_j)\}}{m+l} \quad (1)$$

While $GSim(p, q)$ means the similarity coefficient between service p and service q according to the above section.

3.3 Link-Based Similarity Coefficient

The link-based similarity coefficient of two sets of links L_p and L_q , denoted by *LinkSim* (L_p , L_q), is similar to the Node-based similarity. If we can calculate the similarity of two links, the similarity between two sets of links is same to the above one.

First we analyze the similarity between two links without weights. $\langle S_1, S_2 \rangle$ and $\langle S_3, S_4 \rangle$ are two links, while S_1 , S_2 , S_3 and S_4 are Web services. The similarity between two links is denoted as *LinkSim* ($\langle S_1, S_2 \rangle$, $\langle S_3, S_4 \rangle$). Intuitively, $\langle S_1, S_2 \rangle$ and $\langle S_3, S_4 \rangle$ are similar if S_1 and S_3 are similar, S_2 and S_4 are similar. For example, similarity coefficient between $\langle S_1, S_2 \rangle$ and $\langle S_1, S_2 \rangle$ should be one, and similarity between $\langle S_1, S_2 \rangle$ and $\langle S_2, S_1 \rangle$ will depend on similarity between S_1 and S_2 . The evaluation of the *LinkSim* () coefficient between two links is generally given as follows:

$$\text{LinkSim}(\langle S_1, S_2 \rangle, \langle S_3, S_4 \rangle) = \frac{GSim(S_1, S_3) + GSim(S_2, S_4)}{2} \quad (2)$$

Then we will use the same method described in the above section to calculate the similarity between two sets when the similarity between two elements of sets is known in advance. Because each link has a weight, the difference from Formula 1 is that each maximum is multiplied by two link weights. Then the evaluation of the *LinkSim* () coefficient between two sets of links is given as follows:

$$\text{LinkSim}(L_p, L_q) = \frac{\sum_{i=1}^n w_i^p \cdot w_s^q \cdot \max_{s=1}^k \{LinkSim(Lp_i, Lq_s)\} + \sum_{j=1}^k w_s^p \cdot w_j^q \cdot \max_{s=1}^n \{LinkSim(Lp_s, Lq_j)\}}{n+k} \quad (3)$$

3.4 Similarity Coefficients Between Graphs

Now we will synthesize the above two parts to calculate the graph similarity. The similarity coefficient of two graphs of process designs, P and Q denoted by *Global-*

$Sim(P, Q)$, is the measure of their level of overall similarity computed as the weighted sum of the Node-based and Link-based similarity coefficients as follows:

$$GlobalSim(P, Q) = w_{NodeSim} \cdot NormNodeSim(N_p, N_q) + w_{LinkSim} \cdot NormLinkSim(L_p, L_q) \quad (4)$$

where $NormNodeSim(N_p, N_q)$ and $NormLinkSim(L_p, L_q)$ give respectively the values of $NodeSim(N_p, N_q)$ and $LinkSim(L_p, L_q)$ normalized to the range $[0, 1]$; and where weights $w_{NodeSim}$ and $w_{LinkSim}$, with $w_{NodeSim}, w_{LinkSim} \in [0, 1]$ and $w_{NodeSim} + w_{LinkSim} = 1$, are introduced to assess the relevance of each kind of similarity in computing the similarity coefficients.

The use of weights in $GlobalSim(P, Q)$ is motivated by the need of flexible comparison strategies. For instance, to state that the Node-based similarity and Link-based similarity have the same relevance, we choose $w_{NodeSim}, w_{LinkSim} = 0.5$.

According to the similarity coefficients, processes can be classified in similarity families using the $NodeSim$ and $LinkSim$ coefficients, separately or in combination by means of the $GlobalSim$ coefficient. In particular, similarity thresholds can be set to provide different levels of similarity under different perspectives.

Finally, the algorithm of calculating the similarity between process designs is summarized as follows:

Table 2. Algorithm for calculating process similarity

Begin

- 1) Transform process designs into weighted graphs according to the algorithm in Table 1;
- 2) Calculate the similarity between sets of services according to Formula (1);
- 3) Calculate the similarity between sets of links according to Formula (3);
- 4) Calculate the similarity between graphs based on 2) and 3) according to Formula (4);
- 5) The result of Step (4) is the similarity between process designs.

End

4 Evaluation and Application

The algorithm's performance is $O(MN)$ where M and N are respectively the numbers of services in two processes, since it gets the maximum of similarity between sets of services by examining M services and up to N services for each service according Formula (1). Because the number of activities in a process is general small, the application of this algorithm to large-scale data handling is acceptable.

Since 2002 we have worked on a service composition project called FLAME2008 [13], which aims at providing quick construction and modification of personalized business process for the general customers during the Olympic Games 2008 in Beijing [14]. The implementation of process builder tool with the snapshot in Figure 1 has been finished and described in [2]. The algorithm of this paper can be applied to this project in order to reuse these open-source resources by collecting, clustering and

recommending large-scale processes. Figure 2 is the proposed architecture of the processes management that includes the implementation of the algorithm of this paper. The real line represents the control-flow and the dotted line represents the data-flow.

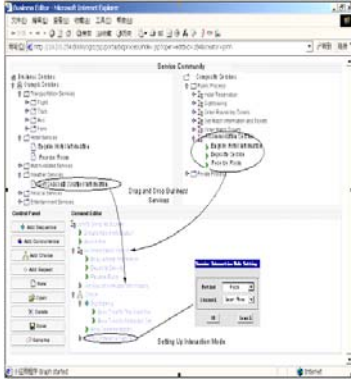


Fig. 1. Snapshot of process builder

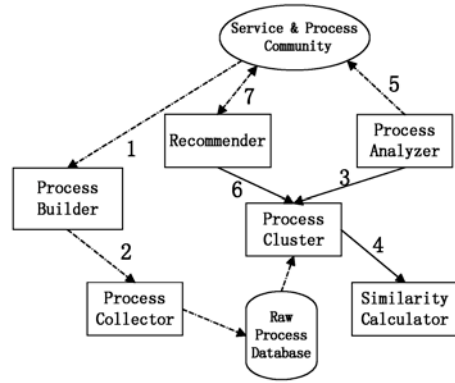


Fig. 2. System Architecture of Process Management

Figure 2 describes the seven steps of managing process designs. Firstly, a third-party agent, such as the Olympiad organizing committee, can define some common-used processes as value-added process designs (see the upper-right corner of Fig.1). Since these pre-defined processes normally cannot 100-percent satisfy the diverse requirements of different users; a user can customize and extend his personalized process design (Step 1 in Fig. 2). The new customized process is shown in the lower part of Fig.1. Secondly, *process collector* will collect these customized processes into raw process database. Thirdly, *process analyzer* will invoke the module *process cluster* to handle raw processes when the number of raw processes becomes big. At that time a *similarity calculator*, the implementation of our algorithm, will be invoked to calculate process similarity. Fifthly, *process analyzer* will analyze users' requirements, extract useful processes and update process community based on the results of clustering. *Recommender* module will also invoke *process cluster* module to cluster data of process community in order to update the recommendation list (Step 6, 7 in Fig. 2).

5 Conclusions

This paper proposes an algorithm for calculating process similarity to collect and cluster open-source processes for future reuses. The contributions of this paper includes: 1) it proposes an algorithm for the first time for calculating the process similarity by their contents to support for clustering processes directly when they are created; and 2) the algorithm of this paper is general and can be applied to every

aspect that refers to the clustering and classifying of large-scale business processes or other structured objects.

The future work includes: 1) the system described in Figure 2 will be implemented in the future and experimental verification of this algorithm will be carried out. 2) For all split cases of an identical XOR-split or AND-split, the weights of links are specified as the average value, which may not always conform to the reality. We will improve on the algorithm in this aspect.

References

1. Wolf-Tilo Balke, Matthias Wagner, Towards Personalized Selection of Web Services. In Proceedings of the 12th International World Wide Web Conference (WWW 2003). Budapest, Hungary, ACM, 2003.
2. Y. Han, H. Geng, etc: VINCA - A Visual and Personalized Business-Level Composition Language for Chaining Web-Based Services. ICSOC 2003: 165-177
3. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G.Schimm, and A.J.M.M. Weijters. Workflow mining: A survey of issues and approaches. Data and Knowledge Engineering, 47(2):237.267, 2003.
4. C.J. van Rijsbergen: Measures of association. In chapter three *automatic classification* of book "*information retrieval*", 1999 <http://www.dcs.gla.ac.uk/~iain/keith/data/pages/38.htm>
5. Gianluigi Greco, Antonella Guzzo, Luigi Pontieri, Domenico Sacca', "Mining Expressive Process Models by Clustering Workflow Traces", Proc. PAKDD 2004, May 26-28, Sydney, Australia, to appear.
6. Linden, G.; Smith, B.; York, J. Amazon.com recommendations: item-to-item collaborative filtering. Internet Computing, IEEE, Volume: 7 Issue: 1, Jan.-Feb., 2003, Page(s): 76 –80.
7. Junji Tomita, Genichiro Kikui: Interactive Web search by graphical query refinement. <http://www10.org/cdrom/posters/1078.pdf>
8. V. De Antonellis, M. Melchiori, and P. Plebani. An Approach to Web Service compatibility in cooperative processes. In Proc. IEEE SAINT2003 of Int.Workshop on Services Oriented Computing: Models, Architectures and Application (SOC2003), Orlando, Florida, USA, 2003.
9. Rolly Intan, Masao Mukaidono: Degree of Similarity in Fuzzy Partition. AFSS International Conference on Fuzzy Systems. Calcutta, India, February 3-6, 2002.
10. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. Distributed and Parallel Databases, 14(3), pages 5-51, July 2003.
11. P. Wohed, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede: Pattern-Based Analysis of BP4WS. QUT Technical report, FIT-TR-2002-04, Queensland University of Technology, Brisbane, 2002.
12. James J. Buckley, Esfandiar Eslami, Thomas Feuring: Fuzzy Mathematics in Economics and Engineering. January 2002
13. B. Holtkamp, R. Gartmann, et al: FLAME2008-Personalized Web Services for the Olympic Games 2008 in Beijing. Proceedings of eChallenges 2003, Bologna, Italy, Oct, 2003
14. Y. Han, et al, CAFISE: An Approach Enabling On-Demand on figuration of Service Grid Applications, Journal of Computer Science and Technology, Vol.18, No.4, 2003

A Study on Semantic Web-Based Credit Evaluation Service*

Jing Fan, Bo Ren, and Jia-Mei Cai

College of Information Engineering, Zhejiang University of Technology
Zhaohui Liuqu, Hangzhou, 310014, P.R. China
fanjing@zjut.edu.cn

Abstract. In order to solve the problem of how to find the proper service desired by the user quickly and accurately from the large group of credit evaluation Web Services, this paper proposes the credit evaluation service based on Semantic Web to implement the share and reuse of credit evaluation service, and satisfy the request of credit evaluation service from different users. First, the architecture of semantic web-based credit evaluation service is introduced. Then the building of Domain Ontology, Web Service description and Web Service discovery, which are the main issues in the semantic web-based credit evaluation service, are discussed in the paper.

Keywords: Semantic Web, Web Service, Credit Evaluation, Ontology

1 Introduction

Semantic Web proposed by Tim Berners-Lee is not a new term now, and it is being a focus in the research of Internet as the next generation of web [1]. Applying the technology of Semantic Web enables the syntax structure and meaning of web content to be expressed in the semantic form, which can be used to make the computer understand and process the web content better, and implement the information sharing and interoperation. The knowledge in the Semantic Web is presented in a hierarchical structure. On the different layer of the hierarchical structure, the knowledge is expressed by XML, RDF, ontology, logic, etc., which makes Semantic Web having richer semantics than the current web.

Credit evaluation to the customers on their situation is the most frequent activity of the enterprise in the business. Most of the enterprises do not have the ability to evaluate credit of the customers accurately and need the support from the specialized evaluation organization. The credit evaluation services based on Web have been developed quickly. With the increase of Web Services, how to find the proper service desired by the user quickly and accurately from the huge group of Web Services becomes the main problem to be solved in the credit evaluation services based on

* The research work in this paper is supported by the project of National 863 High Technology Planning of China (No. 2003AA413320), and the project of Science and Technology Planning of Zhejiang Province (No. 2004C31099).

Web. The key issue is to improve the description ability of the service provider to the service capability, the accuracy and speed of service finding.

This paper proposes the credit evaluation service based on Semantic Web to implement the share and reuse of credit evaluation service, and satisfy the request of credit evaluation service from different users.

2 Architecture

Since the technology of Web Services based on WSDL and UDDI does not make any use of semantic information, it fails to solve the problem of seeking the services based on description of service function, and can not find the best web services according to the service function matching. Having the strong capability of knowledge storage, the Semantic Web [3] aims at the understanding of user request and automatic processing of computer to help the user to locate the proper service faster.

Based on the research of the diversity of credit evaluation service, the difference of service request and the semantic lack of Web Service, this paper apply the technology of Semantic Web to construct credit evaluation service by making full use of the characteristic and advantages of Semantic Web.

The architecture of semantic web-based credit evaluation service is shown in figure 1.

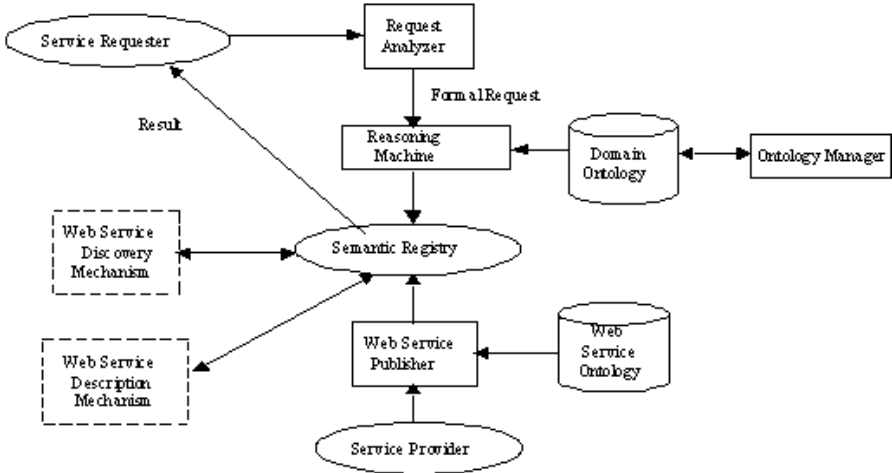


Fig. 1. Architecture of semantic web-based credit evaluation service

The architecture is composed of the following parts:

- Domain ontology. It is built by the service provider and used to represent the common concepts, relations and rules in the domain of credit evaluation.
- Web Service Ontology. It provides a shared representation of concepts, properties of Web Services and is also defined by the service provider. DAML-S and OWL-S can be used to build Web Service Ontology.

- **Ontology Manager.** The providers of credit evaluation services use this module to implement the function management including the creation of a new Domain Ontology, query and maintenance of the existing domain ontology, etc.
- **Request Analyzer.** It is used to analyze the request of service in certain forms, and translate the request into a formalized representation suitable to the domain.
- **Reasoning Machine.** This module is used to reasoning the service request in the formalized representation using Domain Ontology, and construct a semantic service request correspond to the concepts in Domain Ontology.
- **Web Service Publisher.** Service providers use this module to publish the service in registry.
- **Semantic Registry.** It completes the match between the service request and service description using Web Service description mechanism and Web Service discovery mechanism.

According to the above architecture, credit evaluation service works as follows:

- **Service Publication:** the service provider publishes the credit evaluation service in the registry. And in the publication Web Service description mechanism is used to describe credit evaluation service and construct web service ontology.
- **Service Request-Response:** After the service requester submits its request, the analyzer analyzes this request, and translates it into a formalized representation, and then the reasoning machine reasons the request using Domain Ontology. Finally, a proper credit evaluation web service is found and selected with web service discovery mechanism, and the result is returned to the requester.

The next three sections of the paper are focused on the building of Domain Ontology, Web Service description and Web Service discovery.

3 Building Domain Ontology

Building ontology is the key factor to construct semantic web [3]. Ontology has been identified as the basis of semantic annotation and concept sharing. It is comprised of concepts, properties of the concepts, together with the relationships between the concepts. The use of ontology provides the conditions for information sharing and semantic interoperation. By reconstructing queries using ontological concepts in the domain, the semantics in the description of Web Service and query of the requester can be declared explicitly, and semantic web discovery can be achieved through mapping concepts in Web Service descriptions to ontological concepts.

3.1 Method of Building Ontology

There has no general and detailed methodology for building ontology, and based on the analysis of existing methodologies, a five-stage methodology for building ontology in the domain of credit evaluation is proposed. From top to bottom, it can be

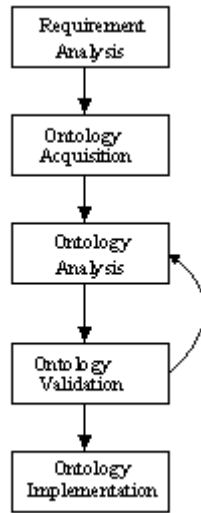


Fig. 2. The five-stage methodology for ontology building

divided into five stages: requirement analysis, ontology acquisition, ontology analysis, ontology validation and ontology implementation (see figure 2).

- Requirement Analysis: determining the purposes, scope, and requirements of ontology in the credit evaluation.
- Ontology Acquisition: identifying the basic concepts related to credit evaluation, and then starting to collect data and information about the concepts.
- Ontology Analysis: analyzing the collected data and information, identifying the basic relations between the basic concepts, and then defining some other concepts derived from the basic concepts, the properties of the derived concepts and the relations among the concepts.
- Ontology Validation: analyzing the concepts identified in Ontology Analysis to avoid redundancy definition of the concepts.
- Ontology Implementation: expressing the ontology using an ontology interpretation language.

And there is a feedback loop between the Ontology Analysis and Ontology Validation. The feedback loop provides the capability to improve the Domain Ontology.

3.2 Organization of Credit Evaluation Ontology

The credit evaluation ontology can be constructed with three abstract classes, Entity Property and Relation. Based on the three abstract classes, a network structure with complex semantic relations and inference functions can be formed through specifying, adding semantic information and axiom definition according the character of credit evaluation.

- Entity. It describes the object or event in the credit evaluation system. The object includes static concepts such as Credit Data and Credit Criteria related to credit evaluation. And the event represents the activities executed on the object. It is the set of dynamic concepts, such as collection of Credit Data, computation on Credit Criteria, etc.
- Property. It describes the properties of Entity. For example, properties of Entity Enterprise-evaluated include name, address, telephone number, etc., and properties of Entity Credit Data include name, source of data and value, etc.
- Relation. It can describe the one-one, one-many, and many-many relations between the entities. Generally, the relations between entities contain not only the normal relations, such as has-part, part-of, and is-a relation, etc., but also the specialized relations for the credit evaluation domain.

In the credit evaluation ontology, there are five types of Entity: Enterprise-evaluated, Credit Data, Credit Criteria, Evaluation Method and Credit Knowledge defined as ontology concepts. Figure 3 shows the five types of Entity and their Relations. The type of Enterprise-evaluated denotes the enterprise being evaluated. The type of Credit Data denotes all the data related to the enterprise credit. The type of Credit Criteria denotes the factor used to evaluate the enterprise. The type of Evaluation Method denotes the method used to execute on the credit criteria to generate the credit evaluation result of the enterprise. The type of Credit Knowledge denotes the knowledge used to guide the evaluation.

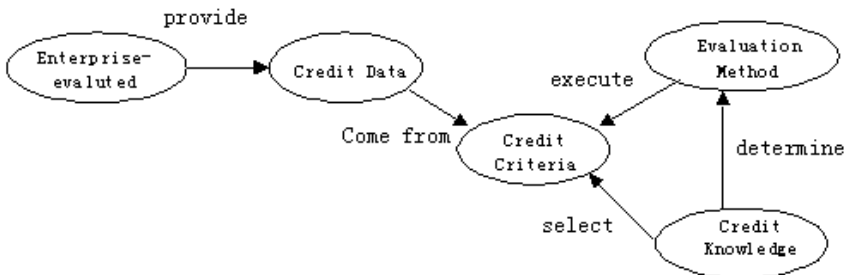


Fig. 3. Ontology concepts and their relations in credit evaluation

3.3 Ontology Modeling Languages

In the area of semantic web, an ontology language should have the following features:

- Compatibility to the syntax character of XML.
- Ability of providing consistent description of concept and information.
- Sufficient ability of inference.
- Compatibility to the existing standard of W3C.

On the comparison among the seven ontology language XOL, SHOE, OML, RDFS, OIL, DAML+OIL and OWL (see Table 1) [5], DAML+OIL and OWL show better than others. Furthermore, DAML+OIL language is more mature than OWL. So DAML+OIL is selected as the ontology modeling language in the research of credit evaluation service based on Semantic Web.

Table 1. The comparison of seven ontology languages (+ indicates supported feature, - indicates unsupported feature)

	XOL	SHOE	OML	RDF(S)	OIL	DAML+OIL	OWL
Partitions	-	-	+	-	+	+	+
Documentation	+	+	+	+	+	+	+
Instance attributes	+	+	+	+	+	+	+
Class attributes	+	-	+	-	+	+	+
Local scope	+	+	+	+	+	+	+
Global scope	+	-	+	+	+	+	+
Default value	+	-	-	-	-	-	-
Type constraints	+	+	+	+	+	+	+
Cardinality constraints	+	-	-	-	+	+	+
Documentation	+	+	+	+	+	+	+
Reasonable	-	+	-	+	+	+	+

4 Web Service Description

Web Service description is the basis of Web Service discovery. Currently, most description specifications of Web Service are based on syntax, such as WSDL. Syntactic description of Web Service is less expressive and can not support intelligent reasoning. In order to support semantic description of Web Service, service description language should be flexible and expressive, and have an ability to express data of semi-structure and constraints.

From the view point of semantic, service description should include three major parts: basic information, capability and other attributes of service. The basic information includes the information of Web Service provider, the site of the WSDL file, and name of the Web Service, etc. Capability description is the key part of Web Service description. It includes input, output, pre-conditions and effects of the Web Service. The description of other attributes provides additive information to help semantic match in service discovery. It includes quality level, response time of service, etc.

DAML-S [6] is a Web Service ontology, which is based on DAML+OIL language. It provides the mark language for describing the aim and usage of a Web Service in unambiguous, computer-interpretable form. DAML-S describes what a service can do besides how it does. It contains three essential types of knowledge about a service: Service Profiles, Service Model, and Service Grounding. Service Profiles defines what the service does. It consists of three types of information: the provider information, the functional description of the service, and a number of features that specify non-functional characteristics of the service. Service Model defines how the service works. It describes the workflow and possible execution paths of the service. Service Grounding specifies the details of how to access a service.

Obviously, DAML-S description of Web Service has richer semantic than the description expressed by WSDL and UDDI. WSDL specification provides the definition and formalization of service query interoperation, but does not provide semantic schema of interoperation. And UDDI only describes the name of service, the tag of service provider, and the access entrance of web service, but does not describe the capability of service. DAML-S focuses on description of the capability of the service, but not the location of the service, and can be used to improve the ability of locating and reasoning and increase the efficiency of Web Service discovery effectively.

The paper uses DAML-S for the description of credit evaluation service, and the concepts in service description are referenced to the concepts in domain ontology.

5 Web Service Discovery

Web Service discovery mechanism is the important part in architecture of Web Service. It enables service requester to search the desired service according to its query and select a best service among search result according to a certain standard. UDDI provides a good environment for the publication, management and maintenance of Web Service, and allows service provider to publish their services in a directory. But it can only provide keyword-based matching and can not support an intelligent search on the higher level. DAML-S can provide a machine-understanding semantic level description of Web Service, which is regarded as the enhancement of WSDL and UDDI. Service discovery based on DAML-S can implement the semantic matching.

After annotating services with semantics, the service provider publishes them in a UDDI registry.

The main work of service publisher module is to transform the DAML-S services to UDDI records through a mapping mechanism. In the mapping mechanism, semantic information about a service provider is mapped to a UDDI businessEntity data structure, and other semantics information of a service such as inputs, outputs, pre-conditions and effects are mapped to tModels in UDDI.

How to match service query with service description is the major challenge in Web Service discovery. Here introduce a three-phase match algorithm. In the first phase, the algorithm finds the proper Web Services that satisfy the category of the desired service. In the second phase, the algorithm checks each Web Service in the set obtained in the first phase for the capability match. In the third phase, the algorithm matches each Web Service in the set obtained in the second phase for the basic information and other attributes of service.

6 Summary

The application of Semantic Web in the credit evaluation services has important significance. This paper describes the main idea of the credit evaluation service based on Semantic Web. The detailed discussion and implementation will be done in the future research work.

References

1. T. Berners-Lee, *Weaving the Web*. Harpur, San Francisco (1999)
2. T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American* (2001)
3. Maedche, S. Staab, "Learning Ontologies for the Semantic Web", *Semantic Web* (2001)
4. M. Uschold, M. King, "Towards a methodology for building ontology", In workshop on basic ontological issues in knowledge sharing, International Joint Conference on Artificial Intelligence (1995) pp: 373-380.
5. Sunci3n G3mez-P3rez and Oscar Corcho, "Ontology Languages for the Semantic Web", In *IEEE Intelligent Systems* (2002) pp: 54-60.
6. N. Ankolekar, M. Burstein, J. Hobbs, et al. "DAML-S: Semantic Markup for Web Services", In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, (2001)
7. M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities", In *ISWC* (2002)
8. C. Bussler, D. Fensel, and A. Maedche, "A Conceptual Architecture for Semantic Web Enabled Web Services SIGMOD Record", *Special Issue Semantic Web and Databases*, (2002)

Research on System Architecture and Service Composition of Spatial Information Grid*

Yu Tang and Ning Jing

School of Electronic Science and Engineering, National University of Defense Technology
Changsha, Hunan, P.R. China
yutang_nudt@163.com, ningjing@nudt.edu.cn

Abstract. Spatial information is a kind of basic and important resource which is needed to be widely shared and applied. But some problems, such as distributed enormous data, heterogeneous data format and system structure, complex processing, and etc, restrict the applications and research of spatial information. Grid can implement large-scale distributed resource sharing, so it provides an effective way to share and integrate spatial information on the web. Based on grid, Web services and OpenGIS specifications, a new service-oriented application grid named Spatial Information Grid (SIG) is proposed. Considering actual application demands, system architecture and service composition are two of the most important research issues of SIG. Then, an open SIG architecture is built, and some key issues of SIG service composition are discussed in detail, i.e. SIG service semigroup, a novel service composition model based on Petri net and graph theory (Service/Resource Net, SRN), and a dynamic service selection model.

1 Introduction

Spatial information is a kind of important information resource that is widely applied in many domains, such as geological survey, census, and so on. Generally, spatial information is defined as any type of information that can be spatially referenced, thus spatial information has three outstanding features different from other types of information, i.e. spatial character, thematic character and temporal character [1].

Corresponding to the three characters, the application process of spatial information (shown in Figure 1) is so complex that it is faced with some problems such as distributed enormous data, heterogeneous system structures, complicated processing, and etc, which are obstacles to realize sharing and integration of spatial information. Traditional techniques can't effectively solve these problems to satisfy increasing demands of spatial information applications. As a new technology to share distributed, sophisticated and heterogeneous resources, grid forms an open and standard information infrastructure to implement large-scale resources sharing [2]. Hence, grid together with Web services [3] and OpenGIS specifications [4] establish the technical

* This work is supported in part by the National High Technology Research and Development 863 Program of China (Grant Nos.2002AA104220, 2002AA131010, 2002AA134010).

foundation of sharing and integration of spatial information. Based on the three key technologies, we propose a new service-oriented application grid named Spatial Information Grid (SIG) [5] aiming to solve those application problems depicted above. The definition of SIG is given as follows:

Definition1: Spatial Information Grid (SIG) is a spatial information infrastructure with the ability of providing services on demands, and implementing organizing, sharing, integration, and collaboration of distributed enormous spatial information.

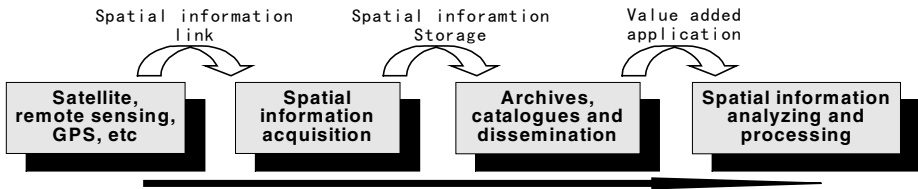


Fig. 1. The application flow of spatial information

SIG is a novel service-oriented framework that defines mechanisms for accessing, managing and exchanging spatial information among entities called SIG services, and enables the integration and composition of services across distributed, dynamic and heterogeneous environment. So service is the technical core of SIG. According to OpenGIS specifications [6], the definition of SIG service is given as below:

Definition 2: SIG service is a collection of spatial operations which is accessible through an XML-based interface and provides spatial application functions.

SIG is a complex spatial information application system, and it involves many research issues among which system architecture forms research foundation of SIG. On the other side, single SIG service can only support simple spatial information application, but most of current applications require wide linking and composition of multiple different SIG services to create new functionality web processes. Thus, service composition becomes a key technology of SIG that needs to be studied firstly. Hence, we put research emphasis on system architecture and service composition here to make sharing and integration of spatial information be easier to be implemented.

The remainder of this paper is organized as follows: In section 2, we introduce system architecture of SIG. Some key issues of service composition are discussed in Section 3. Section 4 presents an application example. Conclusion is given in section 5.

2 System Architecture of SIG

SIG is a powerful system that can be applied through the whole spatial information application course from acquiring, storing to applying. The components of SIG mainly include spatial information acquiring systems, storing systems, processing systems, application systems, multi-layer users, and computing resources (e.g. PCs, servers). These components are linked and integrated by SIG services.

Corresponding to SIG constitution, system architecture describes the structure and framework of SIG. From the view of system theory, SIG system architecture should be studied from two aspects: application architecture and technical architecture.

Considering spatial information application process, we design an open layered application architecture of SIG with seven layers, which is called SIGOA (SIG Open Architecture). The detailed presentation of SIGOA can refer to [5].

Based on SIGOA, we build the technical architecture of SIG (shown in Figure 2) to illustrate technical constitution, contents and framework of SIG, and provide technical basis for designing and implementing SIG applications.

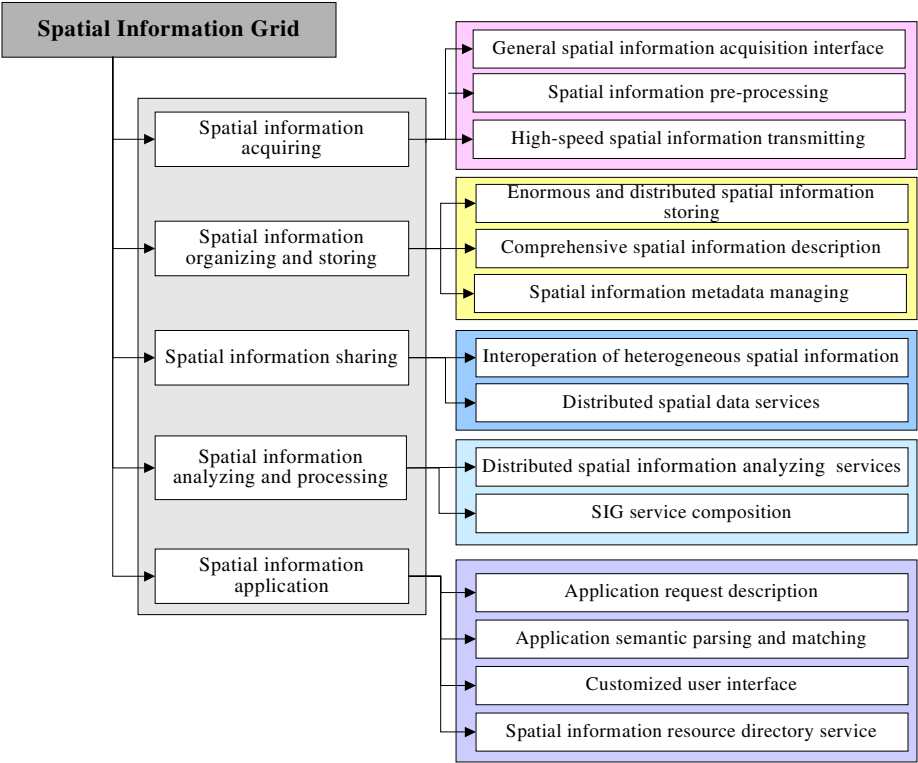


Fig. 2. The technical architecture of SIG

3 SIG Service Composition

SIG is a novel service-oriented spatial information application framework. Single SIG service can only support simple spatial information application, but most of current applications often require wide linking and composition of many different SIG services to built new functionality processes. SIG service composition involves the combination of a number of existing services to produce a more useful service. SIG service composition can be either static (flow logic and invoked services are ap-

pointed before executing service composition) or dynamic (implementing dynamic service discovery, selection, and invocation based on flow logic while executing flow). Then, we propose SIG service composition framework as Figure 3.

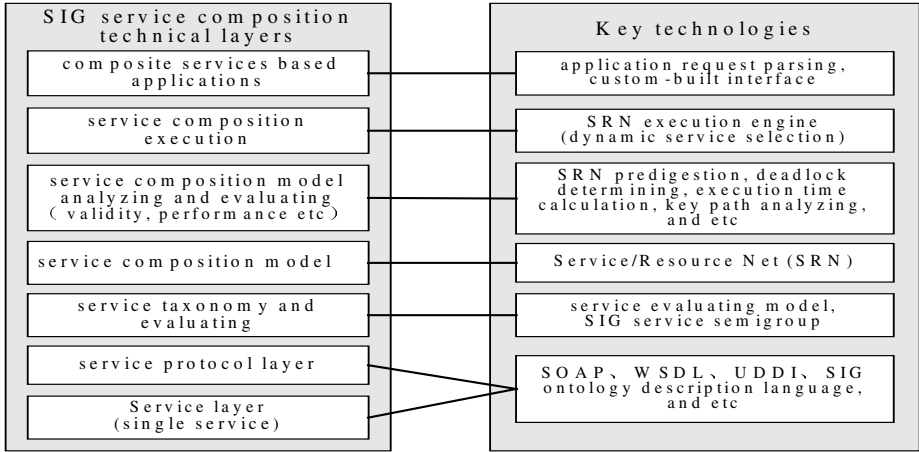


Fig. 3. SIG service composition framework

In accordance with SIG service composition framework and our research foundation, we put research emphasis here on service taxonomy, service composition model, and dynamic service selection model (discussion of service protocol framework and service semantics description can refer to [5]).

3.1 Service Taxonomy Theory – Service Semigroup

Different SIG services can be classified into different service sets. Moreover, different service sets may have similar structure. So it is necessary to study the theories and methods for service taxonomy and service sets similarity determining.

Standard XML-based SIG service protocols and interfaces enable users to link and invoke different SIG services freely. The linking and invoking relation among different SIG services can be regarded as an operation (we define it as *Join*). Based on SIG services sets and *Join* operation, we find that SIG services sets are similar to a semigroup system (basic concepts and theorems of semigroup are in [7]). Thus we propose *SIG Service SemiGroup* (SSSG) as a effective theory for service taxonomy.

Definition 3: *Join* denoted by $\vdash$ is a binary operation which describes the relation that different SIG services invoke each other through standard interfaces.

Definition 4: *SIG Service Semigroup* (SSSG) is a semigroup $(SS, \vdash)$ in which SS is the set of SIG services, i.e. $\forall x, y, z \in SS, (x \vdash y) \vdash z = x \vdash (y \vdash z)$. The symbol $\vdash$ can be omitted, e.g. $(x \vdash y) \vdash z = (xy)z$.

Definition 5: Empty service ($\emptyset$) is a service which has no operation and function.

Definition 6: SIG services monoid $(SS, +, \emptyset)$ is a SIG service semigroup such that $\forall x \in SS, x \emptyset = \emptyset x = x$.

Definition 7: T is a subset of a SSSG $(SS, +)$, if $(T, +)$ is a SSSG, then T is called the *sub-SSSG* of $(SS, +)$.

Definition 8: Given two SSSGs $(SS_1, +)$ and $(SS_2, +)$, map $f: G \rightarrow H$ is a homomorphism if $\forall x, y \in SS_1 \Rightarrow f(x + y) = f(x) + f(y)$; monomorphism, epimorphism, isomorphism and automorphism are four types of homomorphism between SSSGs.

Based on definition 8, we can propose and prove some theorems of determining homomorphism between SSSGs. Further discussion of this point is in another paper.

3.2 Service Composition Model

The process of SIG service composition can be regarded as workflow. As a practical method and tool, Petri net is used widely in modeling workflow, and its basic definitions are in [8,9]. There are at least three good reasons [10] for using Petri net in SIG services composition modeling and analysis:

- Formal semantics despite the graphical nature
- State-based instead of event-based
- Abundance of analysis techniques

But basic Petri net can't accurately describe and model the workflow in which activity and resource flow run synchronously [8,9]. So we introduce three additional elements to extend basic Petri net, i.e. time, conditions, and service taxonomy. Hence we propose a new service composition model named Service/Resource Net (SRN) to effectively and completely describe the process of SIG service composition.

Definition 9: Service/Resource Net (SRN) is an extended Petri net, i.e. a tuple $SRN = (P, T, F, K, CLR, CLS, AC, CN, TM, W, M_0)$, where P is a finite set of places; T is a finite set of transitions; F is a set of flow relation; K is a places capacity function; CLR is a resource taxonomy function; CLS is a services taxonomy function (SSSG is applied here); AC is a flow relation markup function; CN is a condition function on F ; TM is a time function on T ; W is a weight function on F ; M is a marking function (M_0 is the initial marking). Detailed presentation of these elements is in [11].

SRN is a directed bipartite graph with two node types called places and transitions. The nodes are connected via directed arcs. The running of SRN is implemented by firing its transitions, and the basic transition structures of SRN are concluded into six types. Moreover, model analysis and performance evaluating are important research issues of SRN. Further discussion of transition structures, marking firing rules and SRN analyzing methods can refer to [11].

3.3 Dynamic Service Selection Model

The execution process of service composition involves multiple SIG services which are called service nodes. According to process logic and task requirements, the ser-

vice nodes are assigned to implement different functions. In actual applications, there may be multiple service sets that can implement the same function. Furthermore, there may be many services in a service set. So we need select only one service based on selection rules to implement corresponding function of a given service node.

To implement dynamic service selection, we firstly propose a new concept to describe the service set, i.e. Service Family, and its definition is as follows:

Definition 10: Service Family (SF) is a set of SIG services, i.e. $SF = (s_1, s_2, \dots, s_n)$, $s_i (i \in (1, \dots, n))$ is a SIG service. These SIG services are provided by different service providers, but have same invocation interface and can implement same functions.

Considering definitions of SF and SIG ontology [5], we invent a SF selection method based on semantics descriptions of function requests and SIG services to implement matching of service nodes and SFs. This method is to calculate semantic similarity degree of function requests and SFs, and the detailed algorithm is as below:

step1: To generalize the concepts in SIG ontology base, then form semantic vectors of function request and each service of related SF.

step2: To calculate the central vector of each SF, i.e. $d = \sum_{i=1}^n c_i / n$, d denotes central vector of a SF, n is the number of SIG services included in such SF, c_i denotes semantic vector of the i th service in such SF.

step3: To calculate semantic similarity degree of function request vector and central vector of each SF, i.e. $\text{sim}(d_i, d_j)$:

$$\text{sim}(d_i, d_j) = \cos \theta = \frac{\sum_{k=1}^M W_{ik} \times W_{jk}}{\sqrt{(\sum_{k=1}^M W_{ik}^2)(\sum_{k=1}^M W_{jk}^2)}}, \text{ where } d_i \text{ is function request vector, } d_j \text{ is}$$

central vector of the j th SF, M denotes the dimensions of d_i and d_j , W_k denotes the k th dimension of d_i and d_j .

The angle between d_i and d_j is smaller, the $\cos \theta$ value is bigger, and the semantic similarity degree of function request and SF is higher. We will select the SF which makes the value of $\text{sim}(d_i, d_j)$ be maximal as the matching SF of the function request.

After selecting the matching SF, we need to select an optimum service instance from such matching SF to execute appointed function of corresponding service node. Hence, we propose Service Instance Selection Model (SISM) with five selection elements as below:

$$SISM = w_1 \cdot D - w_2 \cdot T - w_3 \cdot C + w_4 \cdot IP + w_5 \cdot R$$

Where w_1, w_2, w_3, w_4, w_5 are corresponding weights, and:

- Degree (D) denotes service grade;
- Time (T) denotes the execution time of service;
- Cost (C) denotes the cost of invoking service, including fee etc;

- Invocation Probability (IP) denotes invocation frequency of service, i.e. for a given $SF = (s_1, s_2, \dots, s_n)$, $\sum_{i=1}^n ip_i = 1, ip_i \in IP$;
- Reliability (R) denotes the reliability of service, and its parameters including the rate of invoking service successfully, maximal load, and etc;

SISM is a complex model composed of above five elements in which the calculation of R is more complicated. The SISM model can be extended conveniently, and we will adjust and modify its parameters according to application demands. In actual execution of SIG service composition, we implement optimum service instance selection by calculating the value of SISM for each Selected SF, and the service instance whose SISM value is maximal will be selected as the optimum service instance, i.e. $os = \{s_i \in SF \mid i \in (1, \dots, n), sism_i = \max(sism_1, sism_2, \dots, sism_n)\}$. The detailed algorithm and rules of SISM will be discussed in our succeeding paper.

4 SIG Application Example

As presented above, SIG can be widely used in most spatial information applications. According to project requirements, we have implemented an experiment system based on SIG services and service composition. A case example of city environment evaluating is illustrated in Figure 4, and its SRN model is shown as Figure 5 (application demo and interfaces of this example are omitted here for space reasons).

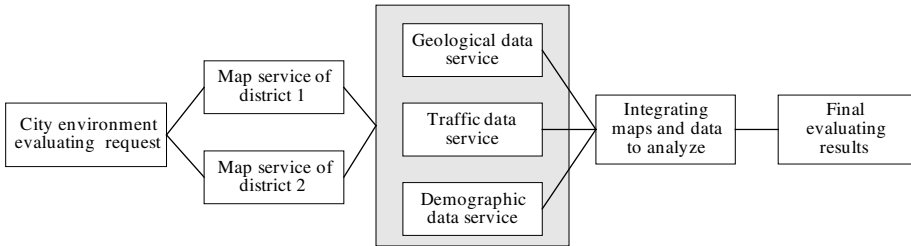
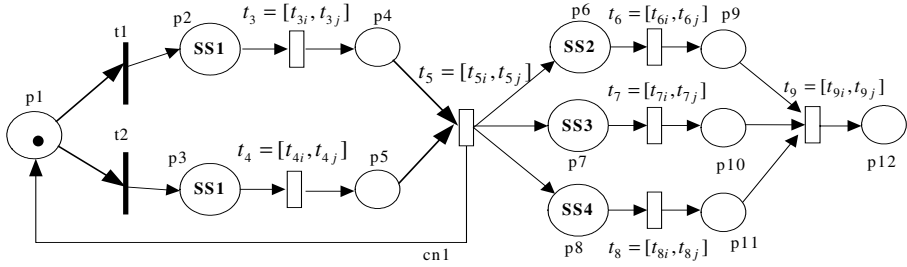


Fig. 4. Application process of city environment evaluating

From this application example, we learn that SIG can implement integration and sharing of distributed heterogeneous spatial information. By composing many SIG services, SIG aggregates spatial information from different distributed departments and organizations all over the city, province, and country to provide powerful abilities of spatial information acquiring, sharing, processing, integration, and applying.

5 Conclusion and Future Work

SIG is a novel service-oriented spatial information infrastructure. The research on system architecture and service composition forms the foundation of SIG research



Transition-- t_1 : request map of district1; t_2 : request map of district2; t_3 : provide map of district1; t_4 : provide map of district2; t_5 : integrate maps; t_6 : provide geological data; t_7 : provide traffic data; t_8 : provide demographic data; t_9 : integrate data to analyze and get final results;
Service taxonomy-- SS₁: map service set; SS₂: geological data service set; SS₃: traffic data service set; SS₄: demographic data service set;
Condition-- cn₁: whether the acquired maps satisfy the application request;

Fig. 5. SRN model of city environment evaluating

and applications. In this paper, we present technical architecture of SIG, propose SIG service composition framework, and discuss some key theories and technologies of SIG service composition, i.e. service taxonomy theory, service composition model, and dynamic service selection model. Furthermore, SIG service semigroup is introduced as a new theory system, a novel service composition model named Service/Resource Net (SRN) is proposed based on Petri net and graph theory, and a dynamic service selection model (SISM) is presented as well.

The research of SIG is in the starting phase. The architecture, concepts, theories and methods in current research need to be perfected and extended. In our future work, we will put research emphasis on enormous spatial data managing, high-speed spatial information transmitting, SIG service composition execution, and etc.

References

1. Longley, P., Goodchild, M., Maguire, D., Rhind, D. (eds.): Geographical Information Systems: Principles, Techniques, Applications and Management. John Wiley & Sons, New York (1999)
2. Foster, I., Kesselman, C. et al: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. (2002). http://www.gridforum.org/ogsi-wg/drafts/ogsa_draft2.9_2002-06-22.pdf
3. World Wide Web Consortium: Web Services Architecture Draft 14. (2002). <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>
4. OpenGIS Consortium: The OpenGIS Abstract Specification Topic 0: Abstract Specification Overview, Version 4.0. (1999). <http://www.opengis.org/docs/99-100r1.pdf>
5. Tang, Y., Chen, L., He, K.T., Jing, N.: SIG: A Service-Oriented Application Grid for Spatial Information Sharing and Integrating. In Proceedings of the 2004 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04), Nevada, USA, CSREA Press (2004) (To appear)

6. OpenGIS Consortium: The OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture, Version 4.3. (2002). <http://www.opengis.org/docs/02-112.pdf>
7. Dai, Y.Q., Hu, G.Z., Chen, W.: Graph Theory and Algebra Structure (in Chinese). Tsinghua University Press, Beijing, China (1999)
8. Murata, T.: Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4) (1989) 541-580
9. Yuan, C.Y.: Petri Net Theory (in Chinese). Publishing House of Electronics Industry, Beijing, China (1998)
10. Van der Aalst, W.M.P.: Petri-net-based Workflow Management Software. In *Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems*, Athens, Georgia (1996) 114-118
11. Tang, Y., Chen, L., He, K.T., Jing, N.: SRN: An Extended Petri-Net-Based Workflow Model for Web Services Composition. In *Proceedings of the 2nd International Conference on Web Services (ICWS 2004)*, San Diego, CA, USA, IEEE Press (2004) (To appear)

Open Language Approach for Dynamic Service Evolution

Thomas Weishäupl and Erich Schikuta

Department of Computer Science and Business Informatics
University of Vienna
Rathausstraße 19/9, A-1010 Vienna, Austria
{thomas.weishaeupl,erich.schikuta}@univie.ac.at

Abstract. This paper introduces a novel approach for dynamic service establishment in a virtual organization. To allow dynamism semantic information has to be processed. A common language is needed. We introduce a novel approach called “Dynamic Service Evolution” (DSE) to establish bilateral agreements in an open process between services and clients. We have applied our approach in N2Grid and present a prototype in this paper. N2Grid is a neural network simulation system exploiting available computing resources in a Grid for neural network specific tasks. By the DSE we solve the problem, that no general standards (languages) for neural network representations exist.

1 Introduction

The Grid started out as a means for sharing resources and was mainly focusing high performance computing. By the integration of Web services as inherent part of the Grid infrastructure the focus evolved to enable collaborations between different virtual organizations or subjects.

A service oriented architecture provides interoperability by defining the interfaces independent of the protocol, according to [1]. Further service semantics are necessary for Grid Services. This means, that service interactions needs not only an agreement over an interface but also over semantics and meaning.

As example can be seen that a user demands storage space of a specific size. He assigns this request to a Grid service using semantic information about the storage space size. A client needs to know about the semantics of a service.

N2Grid [2] is a neural network simulation system exploiting available computing resources for neural network specific tasks. In our system a neural network object is a resource in a Grid. A main problem for using neural networks in a Grid infrastructure is that no standard exists for describing neural networks (problem domain, semantics). Generally the mapping between the problem domain and *service data* of OGSi [3] can not be strictly specified. Therefore we can not implicitly restrict future neural network developments by a specification of service data.

In the upcoming WSRF the service data of OGSi are represented by Resource properties [4]. Resource properties are semantic data of a service described by

XML Schema inside a WSDL interface. This is an advantage over OSGI, but it does not enable dynamic services concerning the semantics, because semantics is a description about the service.

As for N2Grid Services also for other WSRF Services, restricting specifications of possible semantic values is contra productive. Moreover a client needs to get and implement the semantic information of the service in a dynamic way to be Grid aware. A common language (standard) is needed, to get an agreement over the semantics (service description). An open process is needed for the construction of a common language describing the semantic to enable dynamic services.

This paper presents the “Dynamic Service Evolution” (DSE) based on an open language approach to establish dynamic services in a Grid environment. Figure 1 shows DSE’s underlying concepts and technologies. The light-grey boxes denote existing frameworks, white boxes the novel extensions presented in this paper. Further we explain the adaption of the approach and the running prototype in N2Grid.

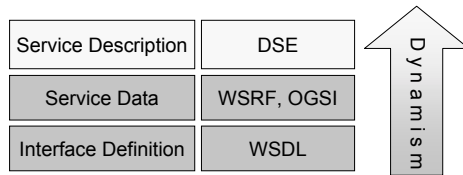


Fig. 1. Dynamic Service Evolution over existing technology

We avoid the further use of the terms Grid services and Web services knowingly, and speak only about services to be independent of any service oriented Grid implementation, as by Globus or “vanilla” Web services.

The paper is structured in three sections. Section 2 presents the novel dynamic services approach. Section 3 shows the adaption of the approach in N2Grid. Finally, Section 4 describes the prototype as a proof-of-concept implementation.

2 Dynamic Service Evolution

As mentioned above, OGSA defines service data to handle semantics in services, as state information, fault and error handling, or other information respectively. These service data describe properties, called resource properties in WSRF. By the service data we can instantiate services and get information about a service. From the client’s point of view there is an inherent disadvantage and interference for dynamic service usage. The semantics of the service data is unknown or at least static.

Only a service and its developer know about the proper semantics. Therefore the service itself must describe its capability and semantics. The service needs to describe its functions or parameter in a client interpretable format, e.g. using a

GUI Meta Description Format in XML Schema. A client can instantiate service data in a dynamic way by using a dynamic GUI. The format of the dynamic service data can be defined in a separate service data schema, as WSRF provides by a Resource Property XML Schema [5].

By an agreement over two XML Schemas, respectively, firstly the client schema to describe the service semantics and secondly the service data schema to describe the service data, the semantics of a service can change without changes in the client implementation. Therefore we have a more powerful, dynamic way to deal with services in a Grid. The XML schema pair builds together a common language.

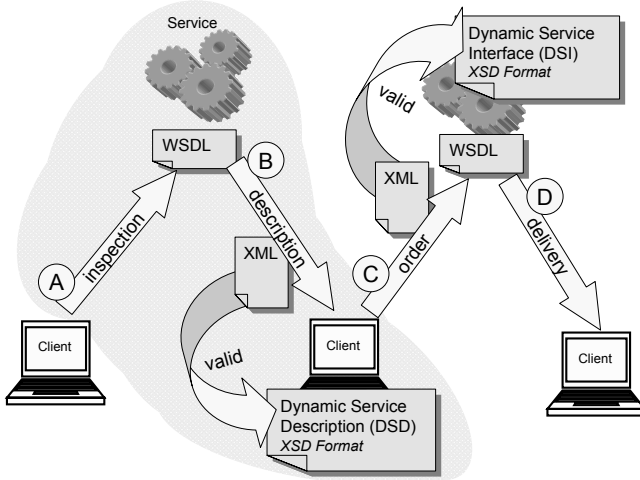


Fig. 2. Dynamic Service Evolution

Figure 2 shows the whole process with the following steps (the new components for the novel approach compared to common Web services is depicted by a light-grey bubble).

- A.** The client contacts a service (found via a registry) to get detailed semantic information about the service. This information is a service description beyond the pure interface description. We name the used format the “Dynamic Service Description” (DSD). It is possible, that the client communicates in the inspection step the preferred service description format, to get a processable response.
- B.** The service sends back the semantics by representing it in a valid client format. This respond can be processes automatically or by a representation on a GUI for user interactions.
- C.** The client produces out of the user input or the automated processing valid service data to get a proper service instance. This service instance can represent resources or other stateful services. We name the service data combined with the interface definition “Dynamic Service Interface” (DSI).

- D.** In this step the service delivers a service instance (resource) or other processing results to the client.

We call this approach “Dynamic Service Evolution” (DSE) because of two reasons.

Firstly, the service can change the semantics dynamically and the appropriate semantical description passes through an evolutionary process. In this case no adaptations are necessary on client side.

Secondly, also the language can go through an evolutionary process. A big advantage of our approach is that no strict standardization of a general and powerful semantic language is necessary. A flexible pair of two schemas defines a new language. Therefore, different languages can be built depending on the problem domain. An open process is possible by independent schema evolutions on client and service side. The definition of a standard is possible by an open process, but dynamic service usage is already enabled without a general standard.

In the following sections we present an application of DSE within the N2Grid problem domain.

3 Dynamic Service Evolution in N2Grid

The N2Grid system is an artificial neural network simulator using the Grid infrastructure as deploying and running environment. It is an evolution of the existing NeuroWeb and NeuroAccess systems. The idea of these systems was to see all components of an artificial neural network as data objects in a database. Now we go ahead and see them as parts of the arising world wide Grid infrastructure. N2Grid is based on a service oriented architecture.

In the N2Grid system we see any neural network as a resource in the Grid. Until now we do not have a strong delimited descriptive language for neural networks to describe the resource. In the future, new paradigms will require new languages. Therefore, we apply our novel dynamic services approach. To prove our approach we give an introduction of the system in the following two subsections and present after this the running prototype.

3.1 N2Grid Use Case

The N2Grid system allows to run the following tasks of neural network simulation remotely in the Grid:

1. Training of neural networks
2. Evaluation of neural networks
3. Processing of data by a neural network
4. Archiving of paradigms
5. Archiving of network objects (nodes, structure, weights, etc.)
6. Using data sources and archives

Task 1, 2 and 3 are integrated into the *Simulation Service* of the N2Grid system, which accomplishes the training, evaluation and propagation function of

the neural network simulator. The necessary data are provided by other N2Grid services, described below.

Task 4 is implemented as N2Grid *Paradigm Archive Service*. Trained neural network objects can be archived for later use.

Task 5 and 6 are unified by the N2Grid *Data Services*. OGSADAI provides the access to a database storing all training-, evaluation-, propagation-data and network objects (nodes, structure, weights, etc.). To provide more flexibility, references (GridFTP URLs) to flat files can be registered in the database which can be accessed directly by the neural network simulation system.

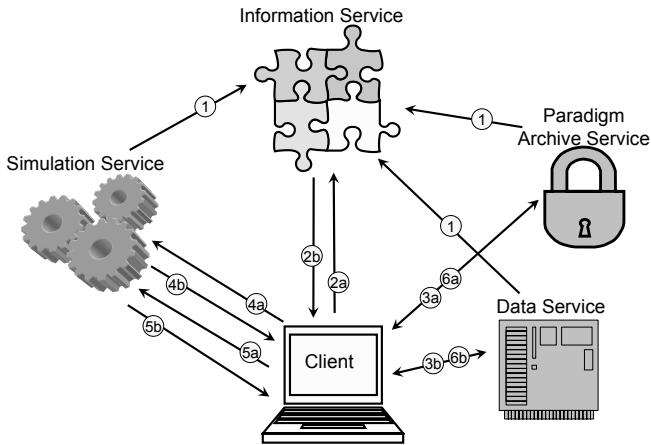


Fig. 3. N2Grid Scenario applied DSE

Figure 3 shows the typical interactions between the N2Grid components during a common N2Grid system usage scenario. It can be described by the following steps:

1. All types of N2Grid services publish their availability in the N2Grid information service.
2. The client queries the information service to discover N2Grid services. These are for example different implementations of one paradigm as back-propagation network or also different paradigms for a specific problem domain.
3. In step 3 the client can compose and search data for a later simulation run. It is also possible to reload a trained network from an archive.
4. Step 4 applies the first two steps of DSE:
 - (a) The client contacts the service to get detailed information about the capability of the neural network simulation service.
 - (b) The service responds by a description, which is representable on the client GUI for user interaction. For example, the user can define the size and structure of network supported by the service.

5. Step 5 applies the second two steps of DSE:
 - (a) The client defines the structure of a new neural network and submits the training data.
 - (b) The service trains the new generated network and returns the result (trained network) to the client.
6. The client archives the result in the intended services (paradigm archive and data service).

In a final release of the N2Grid system the client will only communicate with a broker, which optimizes the access to the resources and delivers the Grid specific transparency. This goes beyond the functionality of an information service or other registries like UDDI.

4 N2Grid Prototype of Dynamic Service Evolution

We applied our dynamic services approach in N2Grid for the interaction between the client and the N2Grid simulation service, which is a proof-of-concept implementation for our novel approach. We need dynamism because of the lack of a general neural network language. The semantics can not be defined strictly.

The implementation is based on the Web service architecture [6] and uses WSDL for the interface definition. We use the Apache Axis Web service container [7] to run our services, inside J2EE runtime environment.

The service publishes its description in the registry (N2Grid information service). It can use the same description also for the semantic description for the client to establish flexibility. The client can search in the registry for a specific property and finds a corresponding simulation service.

The header of the XML Schema for describing the semantics of the N2Grid simulation service is listed below. The N2Grid simulation services can use this schema in a proper way. For example by this schema it publishes the possible parameter and available training method of the implemented neural network algorithm of the service. Later, after further developments of the service, the description can change dynamically.

```
<?xml version="1.0" encoding="UTF-8"?>
...
  <xs:element name="TRAINSERVICE" type="xs:anyURI" />
  <xs:element name="EVALUATIONSERVICE"
    type="xs:anyURI" />
  <xs:element name="STRUCTURE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="INPUT"
          type="BLOCKTYPE" />
        <xs:element name="MAXHIDDENBLOCKS"
          type="SIZEMAX" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
...
```

An example of a concrete N2Grid simulation service description is shown in the following listing. We get information on the available neural network structure and other characteristics of the N2Grid simulation service by the XML document. Based on the information of this document the client can produce dynamically a GUI for user interactions, allowing the user to define a specific neural network.

The GUI can be created out of our service description, but we can also apply a standard GUI language to describe our service, as e.g. XUL [8] from the Mozilla project.

```
<?xml version="1.0" encoding="UTF-8"?>
...
<TRAINSERVICE>http://cs.univie.ac.at/trainservice
  </TRAINSERVICE>
<EVALUATIONSERVICE>http://cs.univie.ac.at/evalservice
  </EVALUATIONSERVICE>
<STRUCTURE>
  <INPUT>
    <ID>input1</ID>
    <DIMMIN>1</DIMMIN>
    <DIMMAX>1</DIMMAX>
    <SIZEMIN>1</SIZEMIN>
    <SIZEMAX>unbounded</SIZEMAX>
  </INPUT>
  <MAXHIDDENBLOCKS>unbounded</MAXHIDDENBLOCKS>
...

```

The definition of a specific neural network is submitted to the N2Grid simulation service by the second XML Schema. This schema defines the service data for OGSi, Resource Properties for WSRF or any other service instance data depending on the service implementation. The following listing shows an example XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
...
<xs:element name="NNDEFINITION">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NNSERVICEID" type="xs:string"/>
      <xs:element name="PARADIGM" type="xs:string"/>
      <xs:element name="DESCRIPTION" type="xs:string"/>
      <xs:element name="STRUCTURE"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
...

```

The two listed XML Schemes define a common language used in our system. A service-client pair has to agree on one schema pair. We learned that the possible dynamism is much more powerful than the usage of ordinary service data only, because of the following reasons:

- A second schema gives the service the possibility to change the semantics inside the service without adaptation on the client side.

- By decoupling two parts of one language, only a smaller part of the system has to be changed or extended in the cases of changes in one schema, or introduction of a new schema.
- The client can implement and interpret different semantic schemas and map them to one common service interface at the same time.

5 Conclusion

We presented a novel dynamic services approach called “Dynamic Service Evolution” (DSE), which we have applied in the N2Grid project. Our approach extends the introduction of “service data” in OGSI and WSRF to handle also the semantics of a service. Two schemas (DSD and DSI) are used to decouple the problem domain (semantics) from the pure interface properties and define a language by an open process. Our approach joins service oriented architectures and real Grids, which have a dynamic environment as key issue. Our approach empowers the community to develop in an open process new languages (standards) to handle semantics. Summed up, we overcome the issues of complex standardization for dynamic environments and provide a flexible evolution of dynamic interactions.

References

1. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum (2002)
2. Schikuta, E., Weishäupl, T.: N2Grid: Neural Networks in the Grid. In: International Joint Conference on Neural Networks 2004 (IJCNN 2004), Budapest, Hungary, IEEE (2004)
3. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., Vanderbilt, P.: Open Grid Services Infrastructure (OGSI) Version 1.0. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum (2003) https://forge.gridforum.org/projects/ogsi-wg/document/Final_OGSI_Specification_V1.0/en/1.
4. Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Maguire, T., Snelling, D., Tuecke, S.: From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution (2004) Version 1.1.
5. Foster, I., Frey, J., Graham, S., Tuecke, S., Czajkowski, K., Ferguson, D., Leymann, F., Nally, M., Sedukhin, I., Snelling, D., Storey, T., Vambenepe, W., Weerawarana, S.: Modeling Stateful Resources with Web Services (2004) Version 1.1.
6. W3C: Web Services Architecture. (2003) W3C Working Draft <http://www.w3.org/TR/ws-arch/>.
7. Apache Project: Apache WebServices Axis Project. (2004) <http://ws.apache.org/axis>.
8. Mozilla.org: XML User Interface Language (XUL). (2004)

Adaptive Grid Workflow Scheduling Algorithm

Shaohua Zhang^{1,2}, Yujin Wu², and Ning Gu²

¹ Shanghai Development Center of Computer Software Technology
Shanghai, China, 200235

² Department of Computing and Information Technology, Fudan University
Shanghai, China, 200433
{shaohuazhang, yujingwu, ninggu}@fudan.edu.cn

Abstract. The scheduling policy and algorithm of grid workflow determine the effectiveness and efficiency of grid workflow tasks, which is the key technology in grid workflow. Based on the definition of grid workflow tasks dynamic ready queue, critical factor, dynamic factor and prior factor, the grid workflow tasks selection algorithm, resource selection algorithm and tasks allocation algorithm are presented, which constitute the workflow dynamic scheduling with multiple policies. It can handle the dynamism of grid resource. Some simulation experiments on the prototype are implemented and analyzed, which show that the algorithm has the advantage of efficiency and practice.

1 Introduction

More and more powerful computing and collaborative grid applications that require tremendous resources are constructed as the Grid researches and Grid infrastructure greatly advance. Many applications are extraordinary complicated which are constrained by temporal and resource relationship. Grid workflow can conveniently construct, execute, manage and monitor grid applications, and automate grid applications with great efficiency. Due to the dynamism, distribution, heterogeneity and autonomy of grid applications, conventional workflow technology can't effectively solve the relative problems of grid environment [1]. So many research groups have presented the specifications and drafts of grid workflow, such as "Grid Workflow" [2] and "GSFL"[3]. A great number of projects have adopted grid workflow component or service to manage effectively grid applications, for example "Gridflow"[4] and "PhyGridN"[5]. The scheduling policy and algorithm of grid workflow are the key technology in grid workflow, which determine the execution and efficiency of grid workflow tasks. The influence of different scheduling policies and algorithms is very different. Because of the temporal and causal relationship, the grid workflow scheduling is much distinct from grid scheduling. Due to the distribution and dynamism, the grid workflow scheduling is more complicate than the traditional workflow scheduling.

Though grid scheduling have been comprehensively studied and analyzed [6,7], and many relative algorithms have been presented, these algorithms mostly take consideration of grid Metatasks that have no dependency between each other. And some works have researched the grid workflow scheduling algorithm based on

GAG(Directly Acyclic Graph) which lack of adaptation varying with performance condition of grid resources[4].

The paper analyzes dependency and constraints, the scheduling phrase and policy of grid task. Based on the definition of grid workflow tasks dynamic ready queue, critical factor, dynamic factor and prior factor, the grid workflow tasks selection algorithm, resource selection algorithm and tasks allocation algorithm are presented, which constitute the workflow dynamic scheduling with multiple polices. Its can handle the dynamism of grid and resource. Some simulation experiments are implemented and analyzed, which show that the algorithm has the advantage of efficiency and practice

2 Grid Workflow Tasks and Scheduling

2.1 Grid Workflow Task Type

Grid workflow task type greatly influences the scheduling and allocation of grid workflow. The constraints of different type tasks are diverse, which lead to different scheduling policy, especially in the grid environment that have not fully central information of resources and tasks. There are two type tasks. One type of tasks are metatasks that are dependent. The other type of tasks are dependent that have temporal or causal relation. Metatasks: The sequence of Metatask execution cannot affect the result of Metatasks because they have not dependent relation. The goal of scheduling algorithm for Metatask is so called Makespan that is an NP problem. Dependent Tasks: The tasks have data, communication, temporal and casual relation. So the order of task execution can not be overturned.

2.2 Scheduling Phase

The workflow scheduling process normally has three phases, which chose the suitable pair of task and resource. It is a NP problem. (1)Matching phase: selecting the resources satisfying the requirements of tasks. The minimal requirements of task resource are defined, which is composed of resource static information, such as the hardware and software architecture, CPU, Memory, bandwidth, organization information. According to the requirements and resources, the resources satisfying the minimal requirements are selected. (2)Scheduling phase: The sequence of tasks executed on resources is determined in the phase. To gain optimal efficiency, the scheduler chose the suitable resource from the resources set according to some constraints and rules. So it is very important to know the dynamic resource and tasks information. Normally some policy must be adopted to get the information and select the pair of task and resource. It is an NP problem, so some heuristics are taken for an near-optimal results. (3)The execution phase: the task is assigned to the chosen resource and is executed. Some management and administration are considered such the cancel, stop, resume, and wait, finish the grid task.

2.3 Scheduling Policy

Static scheduling: the resource is bond on task when the task is modeled in build-time. The task will not executed if the resource is not obtained in the runtime even there are other similar resources that can provide the same function. So the static algorithm often lead to low efficiency, and the task easily fails.

Dynamic scheduling: the resource is not bond on task when the task is modeled in build-time, but the specification of resources. So in the runtime scheduler can select resources to implement the task according to the requirements, and don't worry about the failure of some resources .So normally it is efficient.

Hybrid scheduling: due to the complexity and dynamism of grid. Though the dynamic scheduling are adaptive and efficient. The resources management requires timely information and scheduling system is very complicated. It is very difficult to obtain the full information of grid resource because of the composite method of grid. So some key and special task adopt the static scheduling and other adopt dynamic scheduling.

3 Adaptive Scheduling Algorithm

In this paper, we adopt the D-Petri Net as the grid workflow model language that is extended Color Petri Net. The detail about D-Petri Net is in paper [8], Supposing the grid workflow model is $DP=(P,T:F)$ (For simplicity, We only give the most important elements of DP, P denotes the state set, T denotes the transition set, F denotes the flow set, which are similar with WF-Net[9]), and $P=\{P_1, P_2, \dots, P_m\}$, $T=\{T_1, T_2, \dots, T_n\}$.

Definition 1: Grid workflow critic path.

The sum of all tasks in the critic path is max. It can be inferred from calculation of static expected execution time of the grid workflow instance. The set of task of grid workflow critic path is expressed $TC=\{T_j\}$.The expected max time is TT.

Definition 2: Critic Factor KF. $KF_i = k \frac{TE_i}{TT}$, and $T_i \in TC$, k is a const which adjust the value. The KF_i show the influence to the full grid workflow execution time. Because TC may be changed in the run time of grid workflow instance execution, the KF_i may vary with the TC

Definition 3: Dynamic Factor. $DF_i = k \frac{T_i - TB_i - TE_i}{TT}$, It specify relations between the beginning time and expected beginning time, and the deadline time, and whether the tasks must be adjusted. If the DF_i is positive, the redundant time is more, and if the DF_i is negative, and task has been postponed.

Definition 4: Prior Factor PF. PF_i of grid task specifies integrated priority of the grid work and grid work task. The value has been set at advanced. It can be set according the quality service grid workflow and grid workflow task.

Algorithm 1: Task selection algorithm. Input: grid workflow instance $DP=(P,T:F)$; Output: grid workflow tasks ready queue;

```

Initialize RL= $\Phi$ ;
For every  $t \in T$  do
Count  $KF_t$ ;
Count  $PF_t$ ;
End For;
Select I From P;
Suc= $I^*$ ;
While DP is not finished
For every  $t \in Suc$  do
If t is enabled then
Count  $DF_t$ ;
End If
End For
Repeat //put the every task in Suc into RL according to following rules;
If ( $DF_t < \text{Threshold}$ ) and  $\text{Least}(DF_t)$  Then //Least( $DF_t$ ) denote  $DF_t$  minimal;
Remove t from Suc entering to tail of RL;
Else IF  $\text{Most}(KF_t)$  Then
Remove t from Suc entering to tail of RL;
Else IF  $\text{Most}(PF_t)$  Then
Remove t from Suc entering to tail of RL;
Else  $\text{Earliest}(TB_t)$  Then
Remove t from Suc entering to tail of RL;
End If
Until there is no enabled t in Suc;
If task set Fin in RL have been finished Then
Suc= (Fin *) * U Suc
End IF
End While

```

$$LDF_i = \begin{cases} DF_i > 0.1 : 10 \\ 0.09 < DF_i \leq 0.1 : 9 \\ \dots \\ 0 < DF_i \leq 0.01 : 1 \\ 0 : 0 \\ 0 < DF_i \leq -0.01 : -1 \\ \dots \\ -0.00 < DF_i \leq -0.1 : -9 \\ DF_i < -0.1 : -10 \end{cases}$$

The ready queue maybe varies over the time, which is dependent with the scheduling and grid resources. CF of Some tasks must be recalculated. After the dynamic tasks ready queue is set. The tasks in the queue are ready to schedule and the order of tasks execution is determine according to the constraints of grid resource and grid tasks. The detail of scheduling algorithm is as follows.

Algorithm 2: Scheduling Algorithm. Given the current dynamic tasks ready queue $RL=\{T_1, T_2, \dots, T_n\}$, organizational resources set $OR=\{OR_1, OR_2, \dots, OR_m\}$. In fact the allocation from grid tasks to grid resources is 1:n mapping, which select suitable tasks to suitable resources. In the course of allocation it is most important that considers whether OR satisfies the organization resource, quality of service requirement of the tasks in RL.

Input: the current dynamic tasks ready queue RL, constraints set, organizational resources set OR; Output: The scheduling of the tasks in RL;

```

While DP is not finished
IF RL is  $\Phi$  Then
    Wait();
End if
While RL is not empty
    Select head from RL;
    For I to n do
        If  $OR_i$  satisfied(requirements) then
            Allocated OR to Execute head;
        Else Put the head in the WL;
        Waiting for trigger event resource add or task finished;
    End if
End for
End while

```

Algorithm 3: Resource selecting algorithm. Input: The task T_j , Resources set $OR=\{OR_1, OR_2, \dots, OR_m\}$; Output: The resources list ORT_j satisfying grid task T_j , and the resources in ORT_j are categorized different levels;

```

For i to m do
    If  $OR_i$  satisfied  $T_j$ , Resource Requirements, Organziaiton Role, QoS requirements
Then
     $OR_i \rightarrow ORT_j$ 
    Switch Resource and QoS is higher // for simplification, 10% denotes 0%—
10%, 20% denotes 11%—20%, and so on.
    Case 10%: quality = 0; break; Case 20%: quality = 1; break;
    Case 40%: quality = 2; break; Case 80%: quality = 3; break;
    Case 160%: quality = 4; break; default: 5;
    End switch
    End if
End for;

```

Algorithm 4: Resource allocation algorithm. The grid resources are dynamically allocated according to the grid tasks parameters and the performance of grid resources. Input: The grid task T_j , the resources list ORT_j satisfying T_j with different quality levels; Output: Grid resources allocation;

For grid task T_j , firstly We consider the quality requirements of T_j , the three factors LKF, LDF, PF can be combined into the quality requirement factor $RF=PF*(11-LDF)*(LKF+1)$; So the idea of resources allocations is that the more RF_j of the grid tasks T_j , the more quality resources from ORT_j allocated to the grid task. The detail rules are as follows.

```

If ( $RF_j > \text{Threshold1}$ ) Then //Threshold can be set according to different interval.
{If there is Resource with level 5 Then
Select the resource;
Else Select the Lower Reource; //Select the neighboring low resource.

```

```

End if; }
.....
Else If (RFj>Threshold5 ) Then
{If there is Resource with level 1 Then
Select the resource;
Else Select the Lower Or Higher Reource;
End if; }
Else {If there is Resource with level 0 Then
Select the resource;
Else Select the Higher Reource;
End if; }
End if

```

Monitoring and Re-scheduling

In the runtime of grid workflow instance the grid resources maybe dynamically add or exit. Sometimes the failure of resources lead to that tasks can not be executed after the grid tasks are allocated the resources. The monitoring and scheduling proposed in the paper can resolve the problem that the grid tasks aren't successfully executed or overtime.

4 Algorithm Analysis and Simulation Experiment

The simulated experiments are classified two groups. The first group is the traditional grid workflow scheduling algorithm (DAG), and the second is the adaptive scheduling algorithm presented in this paper.

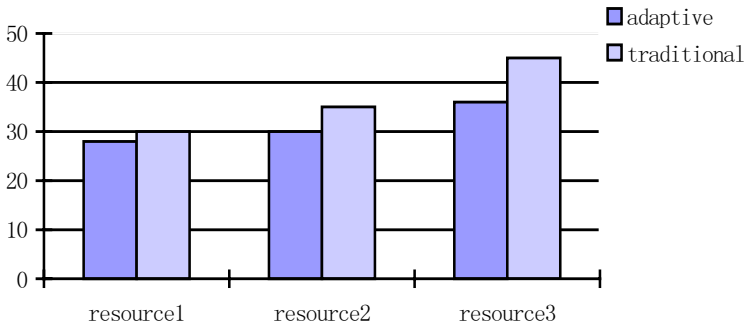


Fig. 1. Simulation experiment Result

The workflow instances comprise of 50 tasks that have dependent relation. Each task has random time between 1-5. In order to simulate the dynamism of grid resources, in the course of grid tasks execution, the grid resources vary. The extension are 20%(resource2), 40%(resource3) respectively. The result is figure1. there are some conclusions from the figure: When the grid resources are static, the execution time is shortest for traditional method and adaptive method. The more varying extension of grid resource, the more execution time. From the simulation condition the decrease of

grid resources affect much the scheduling and execution time. When the varying extension of grid resource is more. The adaptive scheduling method is relative better than traditional method, which is benefited from the factors of the adaptive method, and the Monitoring and Re-scheduling can timely cancel some overtime grid tasks and re-schedule them.

We have developed an grid workflow prototype which consists of user portal, Grid workflow dynamic modeling tool, resource management component, grid services management, performance management, grid workflow engine, grid workflow execution administration. It's very easy to construct the grid workflow model and manage the grid application. The experiment is simulated on the prototype.

5 Related Works

There are a great deal of works studying grid workflow and grid scheduling. Global Grid Forum proposes a standard for the sequencing of complex high-performance computational tasks within a Grid [10]. OGSA defines grid workflow service [11]. Grid Computing Environments (GCE) and the Grid Service Management Frameworks (GSM) Research Groups present a grid workflow management architecture [2]. Grid-Flow[4] includes services of both global grid workflow management and local grid sub-workflow scheduling. Simulation, execution and monitoring functionalities are provided at the global grid level. McRunjob[12] is a grid workflow manager used to manage the generation of large numbers of production processing jobs in High Energy Physics. The project in [13,14] is part of PhyGridN, mainly includes Chimera and Pegasus which are used to create and manage the grid computational workflow that must be present to deal with the challenging application requirements. Chimera allows users and applications to describe data products in terms of abstract workflow and to execute the workflow on the Grid. Though there are substantive scheduling algorithms about the grid metatasks. Such as Min-Min, Max-Min, Suffrage, Genetic algorithm[6]. There is a limited amount of works related to grid workflow scheduling to convert a task scheduling list. The head task is removed and scheduled to the suitable resource. The process is not stopped until all tasks in the list are finished[15].

6 Conclusion and Future Work

Grid workflow scheduling algorithm can effectively and efficiently allocate the grid resources and execution the grid application. Based on the definition of grid workflow tasks dynamic ready queue, critical factor, dynamic factor, prior factor, the grid workflow tasks selection algorithm, resource selection algorithm and tasks allocation algorithm are presented, which constitute the workflow dynamic scheduling with multiple polices. Its can handle the dynamism of grid and resource. The prototype of grid workflow has been developed and some simulation experiments are implemented and analyzed, which show that the algorithm has the advantage of efficiency and practice. Though the simulation experiments have been implemented on the prototype. We hope some practical grid application will be developed and run on the system. So both the grid workflow system and the adaptive algorithm will be further studied.

References

1. I. Foster, C. Kesselman, et al. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications. vol. 15, pp. 200-222, 2001
2. Hugh P. Bivens, Grid Workflow. Grid Computing Environments Working Group, Global Grid Forum, 2001.
3. S. Krishnan, P. Wagstrom, and G. von Laszewski. GSFL: A Workflow Framework for Grid Services. in ANL/MCS-P980-0802, Argonne National Laboratory, 2002
4. J. Cao, S. A. Jarvis, S. Saini, G. R. Nudd. GridFlow: Workflow Management for Grid Computing. In Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03), pp.198-205, 2003.
5. Blythe, J., E. Deelman, et al .The Role of Planning in Grid Computing. in Proc. International Conference on Automated Planning and Scheduling (ICAPS), p. 154-163 2003.
6. HE Xiaoshan, Sun xianhe, Gregor von Laszewski, QoS Guided Min-Min Heuristic for Grid Task Scheduling, J. comput. Sci & Technol. Vol(18), NO(4) July, 2003.
7. Noriyuki Fujimoto, Kenichi Hagihara, A Comparison among Grid Scheduling Algorithms for Independent Coarse-Grained Tasks, Proceedings of the 2004 International Symposium on Applications and the Internet Workshops (SAINTW'04).
8. Shaohua Zhang, Ning Gu, Grid Workflow based on Dynamic Modeling and Scheduling. The 5th International Conference on: Information Technology (ITCC'04)
9. J. Aalst, W.M.P. van der. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 18(3): 21-66
10. Hugh P. Bivens and Judy I. Beiriger. GALE: Grid Access Language for HPC Environments. Available at sass3186.sandia.gov/hpbiven.
11. Ian Foster, C. Kesselman, J. Nick, et al , The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002
12. E. Graham, D. Evans and I. Bertram, McRunjob :A High Energy Physics Workflow Planner for Grid. Computing in High Energy and Nuclear Physics. March 2003, La Jolla, California
13. Blythe, J., E. Deelman, Y. Gil. Planning for workflow construction and maintenance on the grid in ICAPS 03 Workshop on Web Services Composition, pp.8 – 14, 2003.
14. E. Deelman, J. Blythe, et al. Mapping Abstract Complex Workflows onto Grid Environments. Journal of Grid Computing, vol 1, pp 25-29, 2003.
15. Min-You Wu, Wei Shu, Jun Gu, Efficient Local Search for DAG Scheduling, IEEE Transaction and parallel and distributed systems, VOL. 12, NO. 6, JUNE 2001 617-628.

Building a Courseware Grid upon Dart Database Grid

Meiyu Fang¹, Zhaohui Wu², Huajun Chen², Yuxin Mao²,
Guozhou Zheng², Zhao Xu², and Xiaojun Wu²

¹ College of Computer Science, Zhejiang Education Institut
Hangzhou 310033, China
Hwdfmy@hzcnc.com

² Grid Computing Lab, College of Computer Science
Zhejiang University, Hangzhou 310027, China
{wzh, huajunsir, maoyx, zzzgz, Xuzhao}@zju.edu.cn

Abstract. DartGrid is developed by grid computing lab of Zhejiang University, which is built upon Globus toolkit and based on OGSA/WSRF standards, and has been successfully applied in data sharing for Traditional Chinese Medicine in China. In this paper, another application system built on DartGrid for managing courseware resources is introduced. The architecture of this system based on grid is open. The system consists of a series of equal grid-resource nodes. Files of resources need not be uploaded to one node as the c/s server, while the description information of the resources is registered. The salient aspects of this system are: (a) The special courseware ontology was defined as a standard for describing courseware resources; (b) The courseware grid service is to supply semantic registration, semantic query and so on. (c) Its client is a semantic browser.

1 Introduction

In this paper, courseware resources include courseware and other relative material resources during courseware's creation. The courseware resources are becoming richer and richer day by day; however, it leads to many problems such as server overloaded badly, resources reduplication, inconsistent standards, and Resource Island. The key that produces the series of problems lies in the limited share of traditional courseware resources. To the method of description, all kinds of the close special tools and semi-open HTML language are adopted in traditional courseware. It is this method that makes standards inconsistent and information hard to be shared among and inside the courseware resources. To the model of storage and management, The resources are centralized in local network server of C/S mode. It is a limited-level share in local network, because the standards among the servers are not always the same.

Thus it can be seen that the core of the courseware resources' share is:

1. Consistent description language;
2. Open architecture of data sharing on database grid;

Dart database grid supports the two aspects above, DartGrid is developed by grid computing lab of Zhejiang University, which is built upon Globus toolkit and based on OGSA/WSRF standards, and has been successfully applied in data sharing for Traditional Chinese Medicine in China. Another application system built on DartGrid for courseware resources management is introduced. The system consists of a series of equal grid-resource nodes. Resources files need not be uploaded to one node as the c/s server, while the description information of the resources is registered. The salient aspects of this system are:

1. The special courseware ontology was defined as a standard for courseware resources description;
2. The courseware grid service is to supply semantic registration, semantic query and so on;
3. Its client is a semantic courseware resources browser.

2 The Open Management Model of Courseware Resources Based on Grid

The system is composed of the server DartGrid/KG and the client. The client and the server here are different from the traditional C/S client. Nodes of this system (such as node A to N) except DartGrid/KG are clients. The server DartGrid/KG is the central of the Course-grid, which takes charge of the service of resources database and provides resource registration and semantic query but doesn't allocate the resources themselves.

2.1 The Model of the Server DartGrid/KG

What is DartGrid? It is FeiShuo Information Grid, a database grid system based on semantic, which is developed by Grid Computing Lab. Of ZheJiang University. It includes DartGrid/KG[1], Dart-D, DartGrid workflow and so on. Among them, DartGrid/KG, which is the semantic and knowledge core of DartGrid, provide all necessary semantic [2] sustain for DartGrid.

The abstract model of DartGrid/KG is composed of four elemental members. They are semantic browser, knowledge server, ontology server and knowledge base catalog server.

The knowledge server includes the ontology knowledge server, the rule server and the case server and so on. The ontology knowledge server provides basic ontology knowledge service. The rule server and the case server supply not only inference rule and case data relevant to knowledge as supplement of ontology knowledge, but also technique of reasoning based on rule and case. They are used to find knowledge that is not directly described or hidden. They can constitute a virtual organization. They share in the same ontology under which they offer the knowledge service to special field.

The ontology server is used to instruct all shared ontology. The ontology of each virtual organization must come from the ontology server or be in agreement with the server. So the ontology server presents uniform semantic for system DartGrid/KG. It makes semantic communication very easy among all parts. Knowledge catalog server supplies knowledge index service and helps knowledge base node to register, publish, discover and log out etc.

The knowledge server and the ontology server are registered to knowledge catalog server so that the system can find their content and service in time. The knowledge catalog server can also be registered to its higher knowledge catalog server that can supervise all servers under it.

Semantic browser offers such operations as knowledge browse and knowledge query. These element members of the abstract model of DartGrid/KG are also the nodes of DartGrid/KG except for semantic browser. As it is, in knowledge grid DartGrid/KG some nodes take on ontology server, some take on knowledge server, some take on knowledge base catalog server, others even take on two or three roles of them at the same time. The work principle together with organization form and the service acquire measurement of all the servers comply with the same standard OGSA/OGSI. But they provide different services; they are different actors that have different function in local knowledge base grid.

2.2 Client Model

Viewed from client, the logic structure of Course-grid is made up of many sub networks including primary school, middle school, high education school and so on. Each sub network has similar feature. This is a logic database of courseware resources. The architecture is grid. In this system, XML/RDF [3], which can be understood by computer itself, becomes the description language of courseware. Then document during development time can be fully shared. Information can be drawn out and used in other document again.

It is necessary for client to install DartBrowser. DartBrowser is not only the interface of Course-grid server but also the client soft that supplies semantic browse and query operation of courseware at client.

3 Courseware Ontology Building

We must unify the creation and description standards of courseware at first if we want to realize all-sides share and open-ended management. Here a new ontology is built to standardize courseware's description.

We define a super class DartClass at top level for DartGrid/KG, which has an attribute IDproperty. So all the sub classes have an inherited attribute IDproperty that is a key to semantic query.

This is a tree graph of courseware ontology (Fig.1). DartClass is the root of the tree. It has four trunks: PRO, PERSON, CONCERN and RESOURCE.

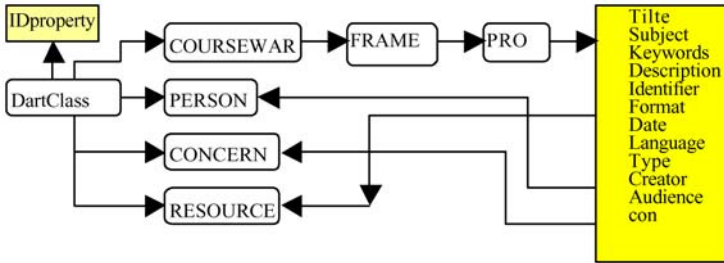


Fig. 1. Tree derivation for courseware ontology

PRO is the Meta Data of courseware resources. It is used to describe content and structure of the courseware resources' information. In addition, there are three class labels for describing the relevant information of courseware. They are PERSON, CONCERN and RESOURCE. In consideration of international standards and national standards, the labels we select not only accurately describe the information of courseware resources but also conform to VCARD, IMS and CELTS42. The core of CELTS42 is in agreement with IMS. The core set of CELTS42 has 11 elements as follows: Title, Subject, Keywords, Description, Identifier, Format, Date, Language, Type, Creator and Audience. The definition and determinant of these elements can also be seen in CELTS42. We define a trunk-node as a class and a leaf-node as property of its trunk-node. PRO has two super class COURSEWARE and FRAME. The class PERSON is used for "Creator", the courseware property. It has three attributes (Name, Address and Email); the class CONCERN is used for the relative resources. It has two attributes (Title and Uri). The RESOURCE class has many sub class (PIC, VIDEO, AUDIO, ANIMATE, COURSEFILE) by course of its format can also be seen in CELTS42. It has three super attributes (Size of, Title, Uri), which will inherit its sub class. Every sub class still has some attributes itself, For Example, TEXT has Font_size and Font_color attributes; ANIMATE has During and Frame_num attributes; AUDIO has During and Volume attributes and so on.

All of the classes and property are banded with XML (Extensible Markup Language [3]). We can use XML to transport, exchange and share data over different platform. XML can express information by means of open-end and well structure. So the courseware's description is innovated. It is open-end and benefit for resources' description and retrieve.

Courseware Ontology has some merits as follows:

1. The class of courseware ontology is clear and definition of the property of the class follows international and national standard without different meanings.
2. It is easy for computer to read and understand.
3. It is fundamental guarantee of open management and necessary condition to share courseware resources.

4 Implementation on DartGrid

The implementation of courseware grid on DartGrid above includes two sides. They are courseware service at server at the top of Fig.2 and courseware browse at client at the bottom of fig.2. At server courseware ontology is defined in classes and properties of class with rdf(s). We import the rdf(s)-ontology to rdf(s) database [4] at knowledge server of DartGrid. At client there are Internet users such as user 1 to user n from whom files about courseware resources are uploaded to database A to database N in c/s server. Through data service publishing, the c/s server becomes a node as node A to node N in CourseGrid. After then node A to N register their databases to the server in CourseGrid. In fact, Relation-ship databases are mapped into semantic databases with rdf(s) through registration. All the registration information about database A to N can be recorded at server. Thus a virtual database is produced. When we browse at client, all records from database A to database N satisfying our query condition can be found. If the Dart Browser that is a soft of client of DartGrid is installed, query will be semantic query and the display of the query result will be semantic browse.

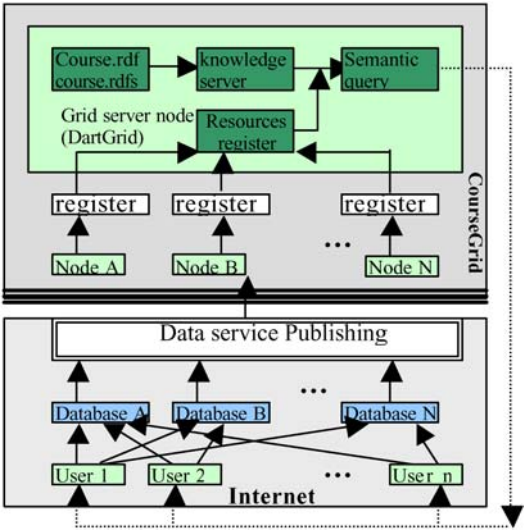


Fig. 2. Workflow of The Open Courseware Resources Management System

4.1 The Courseware Browse at Client

At first, we prepare user database for test. The test resources database is database A and database B corresponding to data node A and data node B of Fig.4. All the data of two databases come from optional courseware collection system from Internet. The model of database is a two-dimension table. There are two tables in each of the database. The table teacher includes the name, the address, and the email of the teachers

who create the courseware. The table main includes fields (Table.1.) as title, content (the abstract), fileurl(the address), course(the subject), idotype(the type) and file-size(file size) of the courseware. Secondly, Service is published to two separate nodes (A and B) of course-grid. Databases are named again, and the physical information including the information about database system, user name, password and data driver information is published to the server.

Table 1. The Mapping from Table Main of Database KJ to courseware ontology

Fields of table main	Attributes of DartClass.COURSEWARE.FRAME.PRO in courseware ontology tree
mainid	Idproperty
fileurl	Fileuri
idoteacher	Person.Idproperty
course	Subject
dateandtime	Date
content	Description
times	
idotype	Type
title	Title
filesize	Resource.sizeof

4.2 The Courseware Service of the Server

The courseware service of the server includes xml/rdf(s) semantic encoding, resources registration and semantic query.

According to the definition of courseware ontology and its feature of document (Fig.4.), we add semantic to courseware ontology in rdf(s) in protég  r-2000 which supports creation of semantic ontology vocabulary, class and instance. We output rdf(s) file from protég  r-2000. The operation is very simple. We create DartClass as a whole super class under which there are classes COURSEWARE, PERSON, CONCERN, and RESOURCE. Each Class has some sub class and some attributes. Pay attention, when two classes have the same attribute, then either children class will inherit the attribute from its father class or the class will add its attribute from the other class through their relation. It is not necessary for us to define a new attribute again. Finally, we output the ontology file named course.rdf(s) from protég  r-2000 using the file menu to save the file in hard driver. Eventually we import it to MySQL database. Then a semantic database can be produced.

DartBrowser is a visual interface designed for courseware registration to Dart-Grid/KG server. It is a mapping from the fields of the database to the semantic ontology. It is imported from file course.rdfs to MsSQL database. (Table.1.). During registration, mapping information of form 1 was written into semantic database and rule database. The resources need not be uploaded. When the user searches the resources, DartGrid/KG will look up the result from the registration table. User will download and browse resources from the data node directly.

Because the information can be drawn out automatically after registration, the basic information about the courseware's general architecture, about the page and the resources list is drawn out to registration center DartGrid/KG. The courseware files was distributed to all over the Grid nodes (node A to node N), so it is different from the C/S model that all resources must be allocated in one server. During using courseware, the center can count the access number of all courseware to get the reference value such as the frequency and the pathway of every courseware. So in this open management system user resources can be published and registered to add to the system any time. Every node of the system can browse and search data of all nodes. From Fig.5, Data is distributed from node A to node N, but to user it is a large semantic logic database whose data comes from node A to node N.

5 Courseware Resources Query

User can search courseware information in DartGrid/KG. Although the information comes from all nodes, it is transparent to users as if the user operates on the same database of the same computer. For example, when a user wants to search courseware of which subject field is “Chinese” from DartGrid/KG. Here the query language is Q3 [1], a database query language like SQL but semantic, it is defined by Grid Computing Lab of ZheJiang University. It is a visual process by mouse clicking to select to write Q3 in DartBrowser automatically. The Q3 language about searching “Chinese” courseware and the query result are as following:

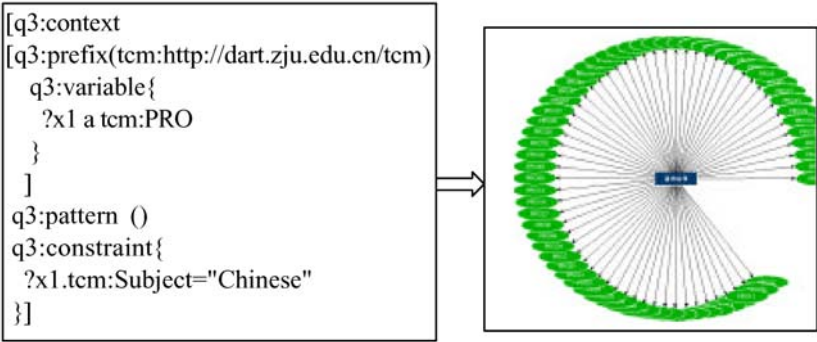


Fig. 3. Instance result of Semantic query

The result records come from all databases of different nodes. So resources share is enhanced than before. In our test case, there are databases KJ1 and KJ2. The result records is as Fig.3. Every oval figures a record.

6 Conclusion and Future Work

In conclusion, DartGrid has been successfully applied in data sharing for Traditional Chinese Medicine in China. At present courseware resources management, as another

application, has been tested successfully. The application of database Grid is becoming very broad. It is used in not only medicine and education but also other areas where there have large numbers of distributed data. But DartGrid is still not perfect. For example, there isn't a whole reference standard of definition of ontology for application area. So there may not be a grid database ontology matching the relation field in database table. And some information of the source database is lost. On the other hand the applications are separate from each other. The repeat same data among the different applications must register again and again. So our future work is that we should insist in researching grid database theory and following up the scent its application. This is important for education because it is possible to manage whole education resources through the open-ended management system on Dart database Grid.

Acknowledgement

This work is supported in part by China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid: Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

References

1. Zhaohui Wu, Huajun Chen, Jiefeng Xu: Knowledge base grid: generic grid architecture for semantic web. 18 (4). Journal of Computer Science and Technology, (2003) 462-473
2. T. Berners-Lee, J. Hendler, and O. Lassila: The Semantic Web. Vol. 284. Scientific American. 5 (2001) 34-43
3. Mark Birbeck, Translated by Jianfeng Pei: Professional XML 2nd Edition. Mechanic Industry Publishing House. (2002) 751-882
4. Farinetti Laura, Bota Florin Mircea, Corno Flulvio. XML based database courseware. 01-05 April. ICDE 2001 - 20th world conference on open learning and distance education. Duesseldorf (Germany)(2001)

A Novel Agent-Based Load Balancing Algorithm for Grid Computing*

Shudong Chen¹, Wenju Zhang¹, Fanyuan Ma¹,
Jianhua Shen², and Minglu Li¹

¹ The Department of Computer Science and Engineering, Shanghai Jiao Tong University
{chenshudong, zwj03, fyma}@sjtu.edu.cn, li-m1@cs.sjtu.edu.cn

² Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences
jhshen@mail.shcnc.ac.cn

Abstract. This paper presents an agent-based resource scheduling algorithm for Grid computing systems. With this scheduling algorithm, the computational resources of the Grid can be dynamically scheduled according to the real-time working load on each node. Thus a Grid system can hold an excellent load balance state. Furthermore, the application of this algorithm is introduced into the practical protein molecules docking applications, which run at the DDG, a Grid computing system for drug discovery and design. Solid experimental results show the load balance and robustness of the proposed algorithm.

1 Introduction

Grid computing technologies provide resource sharing and resource virtualization to end-users, allowing for computational resources to be accessed as a utility. To facilitate such a Grid, a resource management architecture to effectively manage the idle resources is required. Resource management is one of the key research issues for Grid computing. A higher system throughput can be achieved if load balance is added. How to map the computing to the best computational resources and keep a good global load balance is one of the key research areas in the design of a Grid systems. In this paper, an agent-based resources scheduling algorithm will be presented. With this algorithm, a Grid system can hold an excellent load balance state. The application of this algorithm is introduced into the practical protein molecules docking applications, which run at a Grid computing system for drug discovery and design (DDG).

The rest of the paper is organized as follows. Section 2 describes the hybrid resource management architecture of DDG. Section 3 presents the agent-based resources scheduling algorithm and a mathematic model to prove the characteristic of the algorithm. Section 4 analyzes the feasibilities of the algorithm by protein molecule docking experiments. Finally, some concluding remarks are given in Section 5.

* This paper is supported by 973 project (No.2002CB312002) of China, and grand project of the Science and Technology Commission of Shanghai Municipality (No. 03dz15027 and No. 03dz15028).

2 The Working Principle of DDG

There are not only idle cycles available throughout the Internet, but also many users are willing to share their cycles [1] [2]. DDG is a system that allows the end-users to access remote idle resources in a Grid environment. However, resource management architectures of traditional Grid computing systems, such as SETI@home [3] and BOINC [4], almost adopt a master-slave model. This model may bring some frustrating problems, such as single point of failure and performance bottleneck, etc. So, P2P technologies [5] are introduced into DDG. DDG adopts a hybrid Grid-P2P architecture to eliminate problems caused by the centralized resource management models.

Clusters that would like to contribute their idle cycles will join DDG and become the execution nodes. They will be grouped into several Virtual Organizations (VO) by their structures and interests. VOs communicate with each other through the Resource Management Agents in a P2P mode. Depending on their locations, there can be multiple instances of the Resource Management Agent on the P2P network. DDG has a centralized + decentralized P2P network topology. Each local representative Resource Management Agent captures and maintains the resource information of a VO. The agents can exchange information if desired. Figure1 presents the overview of DDG.

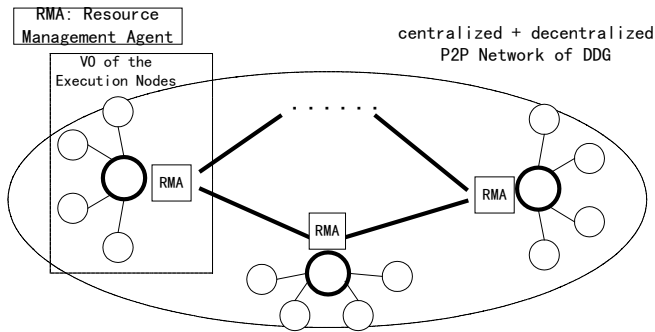


Fig. 1. Hybrid Resource Management Framework of DDG

DDG provides a distributed repository, where execution nodes can publish their real-time workload information. After the scheduler node receives the request, it will query this repository to perform matching of resources in the Grid to satisfy the QoS requirements of the sub-jobs, for example, matching of the hardware requirements of the job. During this step, an agent-based resources scheduling algorithm is used to reach a load balance state of global DDG system.

3 Agent-Based Load Balance

The resource management problems in DDG consists of:

1. Wide-area scheduling of the job requested for idle computational resources into execution nodes in the Grid.
2. Fine-grained resource management on execution nodes during the progress of processing the sub-jobs.

This paper focuses on the first problem of wide-area resource management. VOs have execution nodes located in different places. The distance between the execution nodes can be quite long and nodes also differ in size. The load in one node may be very high while the others may have nothing running on their systems. A higher throughput can be achieved if load balance is added.

3.1 Agent-Based Resource Scheduling Algorithm

The P2P network of execution nodes lacks fixed structures as its nodes come and go freely and the workload of each node varies frequently. It is quite suitable to use agent technology in such a dynamic environment. Agent technology based on P2P has been efficiently used in load balance [6][7]. So an agent-based resources scheduling algorithm has been designed. Considering the hybrid architecture of DDG, the algorithm utilizes the advantages of master-slave structure, P2P technology and agent knowledge.

In DDG, load balance is performed by a large number of agents. Each sub-job is carried by a “mobile” agent. Here, “mobile” means the agent is active and free to choose an Execution node. From this point of view, the sub-jobs are also “mobile”. As a result, each sub-job is considered as an agent. Therefore, we can give a detail description of the dynamic behavior of load balance on DDG.

There is an information repository in DDG which maintains the global information of all the resources in the Grid. The resource-specific information is composed of the static and the dynamic data. Static data mainly contains the Execution node’s processor speed, the number of processors, etc. Dynamic data is the workload and the network delays of the node. The network performance is estimated by the Resource Management Agents through constantly averaging samples of the network delays between the execution nodes. In order to choose the best-suited execution nodes for each incoming jobs to DDG, the algorithm uses computation-specific and resource-specific information. Computation-specific information is mostly included in the end-users request: size in bytes of the input data, computational resources requirement of the job to be processed, and so on.

Moreover, the algorithm in this paper is based on some reasonable assumptions.

Assumption 1. Each agent is free to choose teams in execution nodes and the teams are not.

Assumption 2. It is beneficial for the agents to join short teams and an agent can’t join a team already of the maximum size.

Assumption 3. The number of the agents doesn’t change after the agents are initialized at some given time. This means at a given time there is only one agent to join a team or to leave a team.

There are totally N sub-jobs waiting to be computed in the scheduler node. Available idle computational resources are measured by CPU amount and network bandwidth. If there are m resources units on an Execution node, it means that the CPU amount and network bandwidth of this node is measured as m resource units. We define the maximum size of a team on execution nodes is n . And there is a team T in each Execution node. The following procedure outlines the agent-based resources schedule algorithm.

```

1. Agent-based load balance algorithm {
2.   For each piece of information in the information repository {
3.     if (the idle resource units  $m$  could satisfy the requests of the sub-job) {
4.       make the respective Execution node be an element of the candidate set (CS);
5.       the size of CS becomes  $M+m$ ;
6.     }
7.   }
8.   The scheduler node sends  $\lceil N/M * m \rceil$  agents to each Execution node in CS;
9.   These agents are received by Team  $T$  with size of  $i$ ;
10.  According to the dynamic workload on each Execution node, for each team  $T$  {
11.    if  $i > n$  {
12.      Agents leave the team  $T$  and begin to wander in CS;
13.      The size of team  $T$  becomes  $i-1$ ;
14.      For each wandering agent {
15.        While it encounters a team  $T'$  with size  $j$  in a Execution node {
16.          if ( $(j>n)$  or (it continues to wander))
17.            Continue;
18.          else {
19.            The size of team  $T'$  becomes  $j+1$ ;
20.            Break;
21.          }
22.        }
23.      }
24.    else {
25.      Agents don't leave the team  $T$  and wait to be processed by this Execution node;
26.    }
27.  }
28. }

```

Fig. 2. Pseudocode of the Agent-based Resources Scheduling Algorithm of DDG

3.2 Theoretical Analysis

In order to proof the load balance performance of the agent-based resources scheduling algorithm, a mathematical model has been presented in this paper. Microscopic simulation is an important method for understanding the interactions between agents, so we considered a macroscopic model to evaluation the dynamic behavior of load balance on DDG. The model of coalition formation in [8] describes the interactions between agents with simple local strategies. And it has been used to solve the load balance issues in resources scheduling in a P2P environment [9]. For simplicity, we constructed a model similar to the model of coalition formation because the process of DDG is the opposite one of coalition formation.

Let n denotes the maximum team size, $x_i(t)$ denotes the number of teams of size i at time t , $1 \leq i \leq n$. At the beginning, there are N agents in the Schedule node, $N \geq n$. According to the Assumption 3, there is no net change in the number of agents, then it is expected a realistic dynamic process to conserve the total number of agents in DDG, that is,

$$\sum_{i=1}^n i x_i(t) = N$$

Following model is used to describe how the number of teams of different sizes changes in time:

$$\begin{aligned}
 x_1' &= 2D_2x_2 + \sum_{k=3}^n D_kx_k - 2A_1x_1^2 - x_1 \sum_{k=2}^{n-1} A_kx_k \\
 x_i' &= -D_ix_i + D_{i+1}x_{i+1} + A_{i-1}x_1x_{i-1} - A_ix_1x_i, 2 \leq i \leq n-1 \\
 x_n' &= -D_nx_n + D_{n-1}x_1x_{n-1} \\
 \text{where } \sum_{i=1}^n ix_i &= n, A_j > 0, D_j > 0, x_j \geq 0, 1 \leq j \leq n
 \end{aligned} \tag{1}$$

Here x_i' denotes $dx_i(t)/dt$, which is the rate of change of this number of team of size n at time t . Parameters A_j , the attachment rate, denotes the probability that an agent joins a team. Parameters D_j , the detachment rate, denotes the probability that an agent in a team of size i leaves the team. Parameters A_i and D_i come from agents' experience after they do load balance on the same network for a few times. Model (1) is in agreement with the nature of load balance. In the first equation of (1), " $2D_2x_2$ " shows that one team of size 2 becomes two teams of size 1 after an agents' leaving. " $-2A_1x_1^2$ " shows that two teams of size 1 become one team of size 2 after an agent's joining. For $3 \leq k \leq n$, " D_kx_k " shows that one team of size k becomes one team of size 1 and one team of size $k-1$ after an agent's leaving. For $2 \leq k \leq n-1$, " $-A_kx_1x_k$ " shows that one team of size 1 and one team of size k become one team of size $k+1$ after an agent's joining. Inductively, each expression in (1) can be explained in agreement with the nature of load balance.

It has been proved that there exists a steady state of the dynamic load balance. And the degree of excellence of steady states depends on the probabilities of an agent's actions of joining and wandering. This steady state is a function of x_j :

$$x_i = f(x_1) = c_i x_1^i, 1 \leq i \leq n \text{ where } c_1 = 1, c_i = \frac{A_1 A_2 \dots A_{i-1}}{D_2 D_3 \dots D_i}, 2 \leq i \leq n \tag{2}$$

Equation (2) has obviously shown that if we adjust the values of A_i and D_i , the model can get a good performance. Consider the algorithm, the probability of an agent to join and leave has relation with the team size i and the excellence load number of a Execution node $\lceil N/M * m \rceil$ that is:

$$f(\sum n) = P\left(k * \left| i - \frac{N}{M} * m \right| \right) 1 \leq i \leq n \tag{3}$$

Where k is the adjustment coefficient, this value is defined from the experiences of the agents.

4 Experimental Results

4.1 Experiment Configuration

The feasibility and performance of the algorithm are evaluated by protein molecules docking experiments. The goals of the experiment are to testify the load balance and robustness of the algorithm.

Protein molecules docking experiments are to analyze the similarity between protein molecules provided by end-users and those in biological databases. In our experiments, several databases, such as Specs, ACD, CNPP, NCBI and ACD-SC are used.

Experiments were carried out on a network with four clusters over the Internet and 20 dedicated PCs connected by LAN in a lab of Shanghai Jiaotong University. Each PC had a 1GHz Pentium IV with 256MB of RAM and 40GB of hard disk space and was connected to a 100Mb/s Ethernet LAN. Four clusters are distributed in Shanghai and Beijing: a SGI Origin 3800 cluster of 64 processors and a SUNWAY cluster of 32 processors were deployed at Shanghai Drug Discovery and Design Center, a SUNWAY cluster of 64 processors was deployed at Shanghai High Performance Computing Center, and one SUNWAY cluster of 256 processors was deployed at Beijing Drug Discovery and Design Center.

4.2 Load Balance

In order to ensure the load balance of DDG, the scheduler node needs to avoid assigning jobs to overloaded execution nodes. To maintain good performance, it is important not to exhaust these computational resources. So we hoped that the four super clusters will be allocate more sub-jobs than the PCs. Curves in the Figure 3 show how many computational resources each Execution node has contributed to DDG. The horizontal axis is the observation times every an hour after 0:00am. It can be observed that SGI Origin has provided more computational resources than PCs. The reason is the scheduler node would assign sub-jobs to the SGI more frequently because of its more idle resources. The result was exactly the same with our expectation.

In order to testify the correctness of Equation (3), that is the probability of an agent to join and leave a team could affect the load balance state of DDG, we track the value of A_i , D_i and the adjustment coefficient k . At the time of 5:00, the parameters are $N=3000$, $A_i=0.0001$, $A_i=D_i=9$, $1<i<3000$, $k=0.021$. At the time of 13:00, the parameters are $N=3000$, $A_i=0.0001$, $A_i=D_i=6$, $1<i<3000$, $k=0.045$. Comparing the four curves of those two states, we can find that the load balance state of DDG is different. So, we can draw a conclusion that experimental results have verified the correctness Equation (3). Furthermore, the scheduling algorithm enables DDG to have a good performance at load balance.

4.3 Robustness

In this experiment, we evaluated the ability of the Grid to regain consistency after several execution nodes fail simultaneously. There were 20 PCs with each PC receiv-

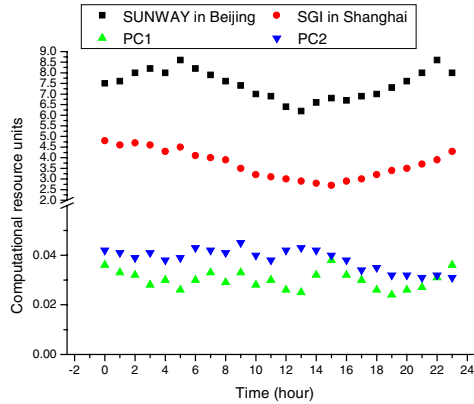


Fig. 3. Load on execution nodes of DDG

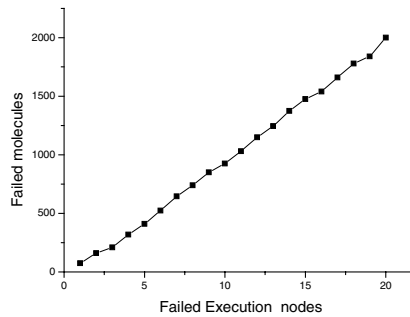


Fig. 4. The Effect of execution nodes Failure on Molecule Docking Experiments

ing 100 protein molecules to dock. Some execution nodes were randomly shut down. After the P2P network stabilized again, we measured the fraction of molecules that could not be processed. Figure 4 shows the effect of Execution node failure on resource scheduling. The molecules docking failure rate was almost equal to nodes failure rate. This was just the fraction of molecules expected to be failure due to the failure of the responsible execution nodes. That is, there was no significant resource scheduling failure in DDG. Thus it can be concluded that DDG is robust in face of the execution nodes' failure.

5 Conclusions

This paper presents an agent-based scheduling algorithm. In the hybrid Grid-P2P resource management framework of DDG, the scheduler node responsible for the central resource scheduling and the execution nodes communicate with each other in a P2P manner to get a load balance state. Utilizing this algorithm, the idle computational resources of DDG can be dynamically scheduled according to the real-time

working load of each execution node. Strict proof has been given that by adjusting the values of A_i , D_i and k , DDG can hold a good performance. Solid experimental results show that DDG can speed up the process of protein molecules docking greatly.

Still, there are some important issues should be explored in the future work. For example, more elements should be concerned in the algorithm other than be confined to only CPUs and network bandwidth.

References

1. D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@home: An experiment in public-resource computing. *Communications of the ACM*, 45: 56-61, 2002.
2. The Intel Philanthropic Peer-to-Peer Program. <http://www.intel.com/cure>.
3. D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Massively distributed computing for SETI. *Computing in Science & Engineering*, Feb. 2001.
4. <http://boinc.berkeley.edu>
5. Andy Oram(ed). *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, 2001.
6. A. Montresor and H. Meling. Messor. Load-balancing through a swarm of autonomous agents. Technical Report, Dept. of Computer Science, University of Bologna, UBLCS-02-08, May 2002.
7. O. Babaoglu, H. Meling, and A. Montresor. Anthill: A framework for the development of agent-based peer-to-peer system. *Proc. Of the 22th Int. Conf. On Distributed Computing Systems*, July 2002.
8. K. Lerman and O. Shehory. Coalition formation for large-scale electronic markets. *Proceedings of the Internal Conference on Multi-Agent Systems, (ICMAS'2000)*, 2000.
9. Yuanshi Wang, Jiming Liu. Macroscopic model of agent-based load balancing on grids. *AAMAS 2003*.

Virtual Battlefield Attack-Defense Countermeasure Simulation on the Grid

Jianhua Yang, Zhaohui Wu, Siliang Tang, and Xiaosheng Guo

Computer Science and Engineering College of Zhejiang University
HangZhou, China, 310027

billyang@mail.hz.zj.cn, wzh@cs.zju.edu.cn,
siliang@tom.com, gxs96213@163.com

Abstract. HLA-based distributed simulation has no mechanisms for managing execution according to the dynamically changing conditions of computing resources and no implementation with dynamical discovery. Grid Computing is designed to coordinate resources that are not subject to centralized control. We present the design of a system supporting execution of HLA-based Parallel and Distributed Simulation (PADS) on the Grid Computing Environment. Firstly, we focus on the architecture of Distributed Interactive Simulation on the Grid (DISG). The architecture definition consists of descriptions of functionality of new tools and Grid Services and indicates where interfaces should be described. We present a model of Virtual Battlefield Attack-Defense Countermeasure Simulation (VBADCS) on the Grid. This model describes how to plug HLA/RTI simulations into the Grid Services framework. Finally, we give an experimental instance of DISG to validate the feasibility and examine the performance of the key techniques of DISG.

1 Introduction

No matter what to be thought of, both martial fields and non-martial ones, system modeling and simulation technique are gradually becoming one of the global research hotspots. From the martial application point of view, the focus of system simulation has recently been on large-scale distributed virtual battlefield attack-defense countermeasure simulation, substitution for local system simulation. Modern military simulation technique has the tendency of the developments towards digitization, virtualization, network, intellectualization, cooperation and synthesis.

In the past decade High Level Architecture (HLA) for Model and Simulation (M&S) was developed as an open, flexible and self-adaptable architecture. Runtime Infrastructure (RTI) is a middleware to realize the Interface Specification of HLA [12,13]. With HLA/RTI to facilitate interoperability among simulation and to promote reuse of simulation middleware, a large-scale distributed simulation can be constructed using a huge number of geographically distributed computing nodes. However, the HLA-Based simulation system does not provide any mechanisms for managing the execution of a simulation according to the dynamically changing conditions of computing resources [10,11].

Grid Computing is aimed to no gap, integrated computing and collaborative circumstance. Grid Computing is to fulfill the flexible, reliable and collaborative re-

source share and solution with the dynamic and complex multi-unit organization [3,5]. Battle simulation based on Grid has the ability to assign large-scale martial simulation mission to distributed environment, to adapt the dynamic network change, to select resource automatically, to adjust running state automatically and to shield the grid malfunction automatically. So we need support for execution of HLA-based distributed interactive simulations in Grid environment [1,8,9].

2 HLA-Based Distributed Simulation on the Grid

Distributed Interactive Simulation on the Grid (DISG) is a specific grid environment to support Advanced Distributive Simulation (ADS) based on HLA entirely, under the background of field simulation, realize the cooperative interaction of large-scale distributed heterogeneous simulation system by synthetically apply ADS, Grid Computing, Virtual Reality (VR), Artificial Intelligence (AI) and other fields knowledge. HLA-based Distributed Interactive Simulation on the Grid has a lot of function and characteristic as the following.

(1) Heterogeneity, Interactivity, Extensibility and Compatibility of DISG

Taking into account the natural heterogeneity of simulation system caused by difference of martial application and division, different functional simulation systems select individual software, hardware, protocol, standard and criterion. Therefore, simulation systems have the specialties as follows: open of simulation architecture; strong interaction on the basis of following uniform standard and criterion; strong extensibility of all sub-systems. Certainly, systems will have to keep good compatibility with existing system. The execution flat of simulation process is HLA/RTI, so DISG supports distributed simulation based on HLA/RTI sufficiently.

(2) Services Dynamic Distributed Registry, Search and Discovery

The Grid Computing Environment (GCE) emphasizes the shared physical resources and services supported by those resources. The services of Open Grid Service Architecture (OGSA) include all kinds of resources [6,7]. The services of Distributed Interactive Simulation on the Grid (DISG) are simulation entity or simulation federate, which has specific application. The simulation entity has strong distribution on time and geography. The simulation entity can dynamically create federation, join federation and quit federation. So DISG must offer high performance and dynamic distributed services registry, search and discovery function.

(3) Resources Dynamical Allocate, Schedule, Optimal and Share

With the function of resource dynamic allocation of Grid Computing the simulation system can enhance the efficiency of system execution and resource utilization. DISG can be self-adaptive to dynamic change of Grid environment, automatically select resource, submit executable code and running data, adjust running state and shield the malfunction of Grid, realize VV&A of the whole life-cycle of simulation. When Grid node quit and terminate simulation task that DISG can transfer the processes running on the node to other nodes, automatically track and adjust the execution states to attain the best efficiency and performance and ensure more strong fault-toleration ability and robustness.

(4) Uniform and Normative Data Format, Flexible and Efficient Communication Strategy

Standard and criterion are the kernel of the simulation. Distributed Interactive Simulation on the Grid (DISG) combines together all the heterogeneous simulation systems dispersed around the world by the network. Therefore, all kinds of standards, protocols and evaluation criterion will have to be standardized uniformly in order to avoid the inconsistency between the heterogeneous systems and conveniently migrate and uniformly manage. The criterion is concerning architecture of model, virtual environment, interaction information, simulation performance and result evaluation.

DISG chooses high efficiency and flexible data communication structure and communication protocol and route strategy because the large-scale distributed simulation system has plenty of data exchange and strict time limit.

(5) Offer Support Tools of Field Simulation

DISG offers many kinds of reusable M&S tools of field application simulation to design, exploit, analyze, execute and evaluate of the simulation result. DISG well supports whole life cycle of simulation and rapid construction of complex simulation system. DISG has user-oriented friendly visual display environment.

3 DISG Architecture

Open Grid Services Architecture (OGSA) defines the concept of Grid Service. It adopts the uniform framework of Web Service. It establishes the foundation of development and realization of distributed interactive simulation on the Grid. It has well solved a lot of mechanism of dynamic service discovery, service establish, whole life-cycle manage and service notification [14,15]. The layered architecture of advanced distributed simulation on the Grid environment mainly works over the layered principle, interrelationship, interface and realization detail.

DISG integrates the Grid Computing Technology to the Distributed Simulation Environment. It takes into account how reasonably and availably to use the Grid Services and Resources on the application layer based on HLA/RTI [1,2,8,9,14,15]. DISG encapsulates distributed simulation environment and all kinds of field simulation middleware on the Grid Services Layer. DISG consists of several layers: Grid Infrastructure Layer (GFL), Grid Service Layer (GSL), Simulation Grid Middleware Layer (SGML), Simulation Application Layer (SAL) as in figure 1. The layered architecture provides modularity and extensibility by each layer interacting with each other using the uniform interfaces.

(1) Grid Infrastructure Layer mainly offers resources on the physical layer, including Simulation Model Resource, Virtual Environment Resource, Computing Resource, Storage Resource, Data Resource, Information Resource, Knowledge Resource, Sensors and Other Equipments. The user of DISG can register and log in the simulation grid environment to use all kinds of resources.

(2) Grid Services Layer consist of three section: Grid Base Services such as Policy, Grid-FTP and GRAM; Grid Core Services such as Discovery, Notification, Registry, Factory and WS-Security; Grid Collective Services such as MDS, RLS, Broker, CAS and User Interaction Services [4,6,7].

(3) Simulation Grid Middleware Layer is the facilities to support collaborative simulation, such as Time Manage, Data Manage, Communication Manage, Analysis and Replay, Scenario Editor, VV&A, Execution and Monitor. Simulation Grid Middleware is the supportive tool of simulation application. It offers users some fundamental services and integrates heterogeneous simulation systems using the HLA/RTI service interface.

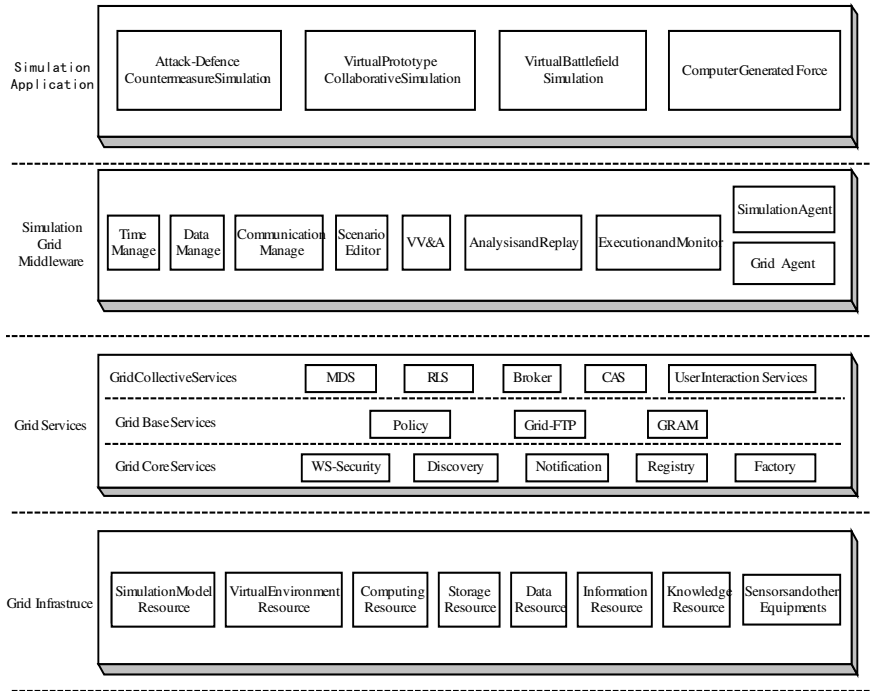


Fig. 1. General Overview of the Layer Architecture of the DISG

(4) Simulation Application Layer supports cooperative design, exploitation, test, execution and evaluation of distributive, virtual, dynamic, heterogeneous complex large-scale simulation systems, such as the Virtual Prototype Collaborative Environment, Distributed Virtual Battlefield Environment, Attack-Defense Countermeasure Simulation and Virtual Computer Generated Force.

Simulation Grid Middleware layer is an intermediate layer between Simulation Application layer and Grid Generic Services layer which is in the charge of a bridge between them in order to support Distributed Simulation Environment (DSE) over Grid Computing Environment (GCE). It is composed of Grid Agent (GA) and Simulation Agent (SA), which allow DSE and GCE to harmoniously interact with each other.

4 Virtual Battlefield Attack-Defense Countermeasure Simulation Model on the Grid

Virtual Battlefield Attack-Defense Countermeasure Simulation (VBADCS) based on Grid is large-scale real-time simulation system under perplexing distributed virtual interaction conditions, which has the advantages of scene division, resource management, information service, register service, system monitor and fault toleration like Grid. System can be self adaptive to dynamic change of Grid, automatically select resource, submit executable code and running data, adjust running state and shield the malfunction of Grid [17]. We present an attack-defense countermeasure simulation model based on Grid, shown as Figure 2.

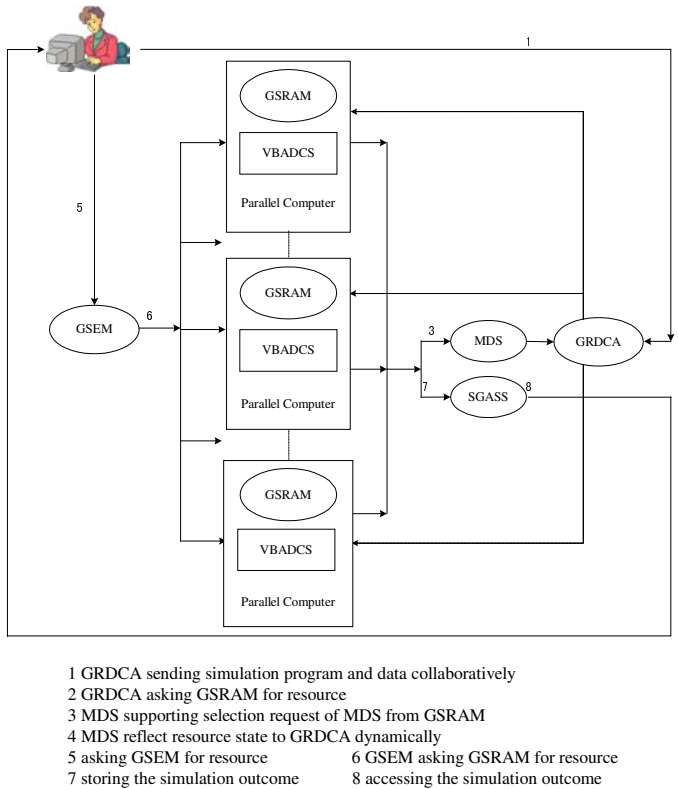


Fig. 2. Attack-Defense Countermeasure Simulation model on the Grid

Under the Grid conditions, each parallel computer is fixed GSRAM (Grid System Resource Allocation Manager) in. GSRAM deal with the local resource and reflect the dynamic state of resource to MDS (Metacomputing Directory Service). MDS can timely map the dynamic change of Grid resource to allocate the Grid resource as the resource allocation table of operation system does. GRDCA (Grid Resource Dynami-

cally Collaborative Allocator) with all the GSRAMs collaboratively allocates resource considering the allocation state.

After asking for enough resource, we make use of the GSEM (Grid System Execution Manager) to transmit executable code of VBADCS and original data automatically to each simulation node and set the different parameters. Simulation outcome and running log can be conveyed to appointed computer by the SGASS (System Global Access Secondary Storage), which makes it convenient to make real-time surveillance and adjustment of simulation state in order to make a distributed real-time display of VBADCS sceneries.

Simulation Application can dynamically adjust running state when working because of the dynamic updating of MDS. When performance of special computer comes down, simulation application can transmit some allocated mission to other computers to keep the system running properly.

5 Implementation: GPS Attack-Defense Countermeasure Simulation on the Grid

GPS Attack-Defense Countermeasure Simulation system is an experimental instance of Distributed Interactive Simulation on the Grid (DISG) to sufficiently validate the feasibility and examine the performance of the critical techniques of DISG as in figure 3. The simulation system has five federates: Red System, Blue System, White System, 3D Scene Display and 2D Plan View Display. Using the Distributive Service Registry, Service Search and Dynamic Resource Allocation Mechanism of OGSA the simulation federate automatically logs on the Simulation Grid Environment to dynamically allocate resource, automatically make notes of simulation data, storage simulation result and VV&A.

The experimental instance consists of Simulation Client, User Interface and Simulation Server. Simulation Client is a simulation entity running on the flat of HLA/RTI to dynamically allocate the most appropriate resources using Grid Search Service as in figure 4. User Interface is the interactive interface between users and Simulation Grid Environment. The users can look over real-time states of simulation system and replay it. Simulation Server consists of all kinds of application models and tools, such as GPS constellation model, GPS receiver model and GPS interferer model, which can be called by simulation clients respectively.

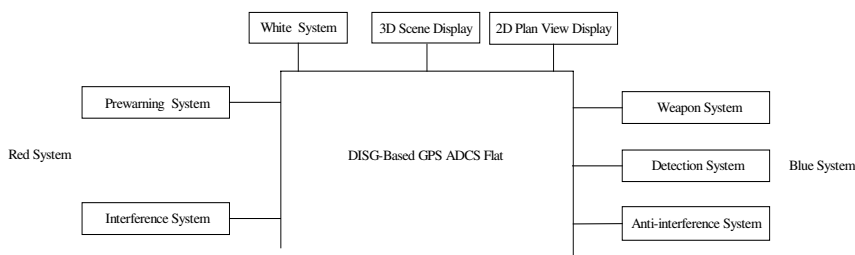


Fig. 3. The Framework of GPS Attack-Defense Countermeasure Simulation Based on DISG

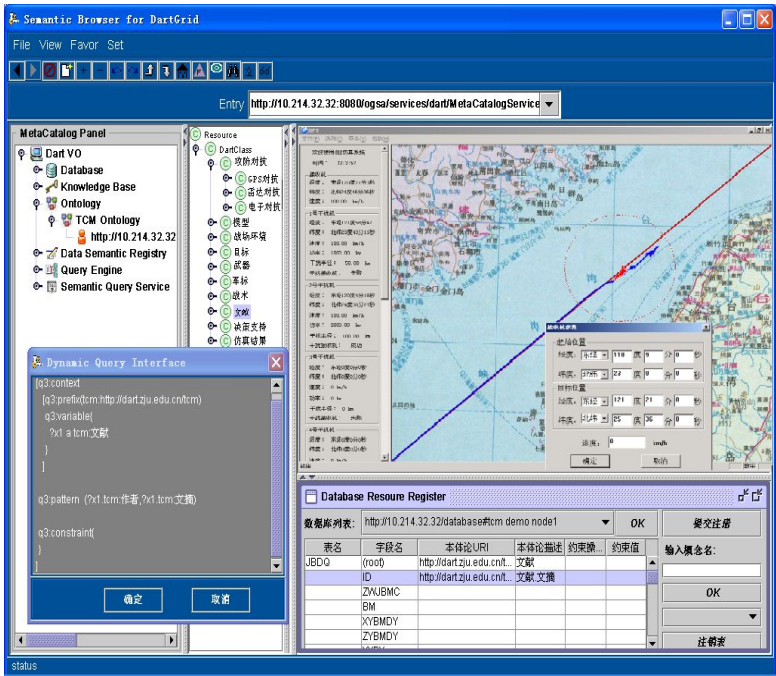


Fig. 4. GPS Attack-Defense Countermeasure Simulation System on DISG

6 Summaries and Future Work

DISG is a large-scale simulation system realized under the distributed virtual circumstance. Simulation Management on the Grid runs through the whole process of the simulation. Military Simulation Management on the Grid relates to the layout, settings, schedule and execution of drilling, that is to say, supposed management and post-analysis. How to take Scenario Management of complex simulation is one of the research tendencies in the future. Techniques on the Grid Computing have development so rapidly that collaborative simulation combined with Grid and Web Service has become one of the development trends of the future simulation techniques [16].

Future VBADCS is the architecture supporting heterogeneous, dynamically extensible and open virtual environment. It will have the capability to deal with large-scale distributed virtual environment, to share the scenery data, to update, to optimize the transmission delay and compensation, to make interaction and cooperation, to make real-time generation, synchronous synthesis of apperception information under perplexing virtual conditions and to solve the mapping and management of indirect perceptive information so that it will ultimately become self-adaptive attack-defense countermeasure simulation system with advanced Artificial Intelligence technologies.

Acknowledgments

The authors acknowledge the support of the Defense Advanced Research Project of the General Armaments Department of the PLA (grant 413150804); China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid: Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

References

1. Zajac K., Tirado-Ramos A., Zhao Z.M., Sloot P., and Bubak M.: Grid Services for HLA-Based Distributed Simulation Frameworks, Across Grids 2003, LNCS 2970, pp. 147–154, 2004. Springer-Verlag Berlin
2. Bubak M., Malawski M., Zajac K.: Towards the CrossGrid Architecture. In: D.Kranzlmeller, P. Kacsuk, J. Dongarra, J. Volker (Eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October 2002, LNCS 2474, pp.16-24. 2002, Springer-Verlag, Berlin.
3. Foster I.: What is the Grid? A three checkpoints list. Grid Today Daily News And Information For The Global Grid Community July 22, 2002: VOL. 1 NO. 6
4. Balis, B., Bubak, M., Funika, W., Szeplieniec, T., and Wismuller, R.: An Infrastructure for Grid Application Monitoring. In Proc. 9th EuroPVM/MPI Users'Group Meeting, Linz, Austria, Sept. 2002, LNCS, pp.41-49, 2002, Springer-Verlag Berlin
5. Foster I., Kesselman C., Tuecke S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International J. Supercomputer Applications, 15(3), 2001
6. Foster I., Kesselman C., Tsudik G., Tuecke S.: A Security Architecture for Computational Grids, Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83–92, 1998.
7. Czajkowski K., Fitzgerald S., Foster I., Kesselman C.: Grid Information Services for Distributed Resource Sharing, Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing, IEEE Press, pp.181-194, August 2001.
8. Kim C.H., Lee T.D., Hwang S.C., and Jeong C.S.: Grid-Based Parallel and Distributed Simulation Environment, PaCT 2003, LNCS 2763, pp.503-508, 2003, Springer-Verlag, Berlin
9. Zajac K., Bubak M., Malawski M., and Sloot P.: Execution and Migration Management of HLA-Based Interactive Simulations on the Grid, PPAM 2003, LNCS 3019, pp. 872–879, 2004. Springer-Verlag Berlin
10. Cai W., Turner S.J., Zhao H.: A Load Management System for Running HLA-based Distributed Simulations over the Grid” in : Proceedings of the 6th IEEE International Workshop on Distributed Simulation and Real-Time Applications 2002, IEEE Press, pp.7 -14, October 2002.
11. Tirado-Ramos A., Zajac K., Zhao Z.M, Sloot P.M.A., Albada D.V., and Bubak M.: Experimental Grid Access for Dynamic Discovery and Data Transfer in Distributed Interactive Simulation Systems, ICCS 2003, LNCS 2657, pp. 284–292, 2003. Springer-Verlag Berlin
12. The HLA Working Group, <http://www.sisostds.org/stdsdev/hla>

13. High Level Architecture Run-Time Infrastructure RTI 1.3-Next Generation Programmer's Guide. <https://www.dmsi.mil/public/transition/hla>
14. Bubak M., Malawski M., and Zajac K.: Architecture of the Grid for Interactive Applications, ICCS 2003, LNCS 2657, pp. 207–213, 2003. Springer-Verlag Berlin
15. Zajac K., Bubak M., Malawski M.: Towards a Grid Management System for HLA-based Interactive Simulations, Proceedings of the Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications 2003, IEEE Press, pp.4 -11, October 2003.
16. Holmes V.P., Johnson W.R., Miller D.J.: Integrating Web Service and Grid Enabling Technologies to Provide Desktop Access to High-Performance Cluster-Based Components for Large-Scale Data Services, Proceedings of the 36th Annual Simulation Symposium 2003, IEEE Press, pp.167 -174, April 2003.
17. Sharon Brunett, Steven Fitzgerald. Metacomputing Supports Large-Scale Distributed Simulations. <http://www.cacr.Caltech.edu/SFExpress>

Virtual Semantic Resource Routing Algorithm for Multimedia Information Grid*

Haiyang Zhang and Huadong Ma

School of Computer Science & Technology
Beijing University of Posts and Telecommunications, Beijing 100876, China
mhd@bupt.edu.cn

Abstract. This paper introduces a virtual hierarchical semantic framework for multimedia information grid according to the semantic relativity between multimedia information resources(MIRs) in semantic web. This framework includes the MIR semantic description model and organization model. On the basis of this framework, we propose the Virtual Semantic Resource Routing (VSRR) algorithm to improve the efficiency of searching MIR and provide the highly precise MIR locating. In this algorithm, we search MIR in multimedia information grid by routing among virtual semantic resource sub-grid according to the semantic relativity of the MIR. Finally, this paper analyzes the performance of VSRR algorithm.

1 Introduction

Information grid, one of branches of grid[1, 2], focuses on effectively integrating of large-scale information resources[3, 4], so that dynamic organization of disparate individuals and/or institutions may collaborate to achieve a shared goal.

At present, the Multimedia Information Grid (MIG) has become the significant application field of information grid. In the MIG, Multimedia Information Resources (MIRs) originating from different derivations have various formats, so we can not exactly locate the requested MIRs and efficiently obtain the aggregate of requested MIRs without semantic description of MIRs.

Sharing MIRs in grid, we need consistent understanding about the data structure, syntax and semantic of MIRs to high-efficiently and precisely search and locate them. Hence the semantic grid seeks to incorporate the semantic web approach into the ongoing grid. Using semantic and ontology in grids can offer high-level support for managing grid resources and designing complex applications that will benefit from the use of semantics [5, 6]. For example, the UK Core e-Science program started its semantic grid initiative, aiming to integrate and bridge the efforts made in the grid and semantic web communities.

Service mode of MIG is the special model in which many servers synchronously provide a service to a user or some users, since the multimedia services require

* The work reported in this paper is partly supported by and the NSFC under Grant 60242002 and the Chinese EYTP of MOE.

strong capability which hardly sufficed permanently by one server. So we not only need precisely locate the requested MIRs, but also need frequently obtain an aggregate of the relative resources. Accordingly, the previous resource discovery policies in network or grids have not perfectly integrated the semantic information and resource routing policy [9, 10], so they are not very suitable for this situation, therefore we propose a Virtual Semantic Resource Routing (VSRR) algorithm which not only integrates the resource semantic information and resource routing policy, but also considers the features of multimedia services with the enormous quantity, a great deal of format in synonymy and extensive distribution of MIRs. In this algorithm, the relative MIRs are linked with semantic beforehand, and so we can rapidly locate an aggregate of relative requested resources. Evidently, we must construct a new grid framework to adapt to the VSRR policy. In this framework, we should describe MIRs by semantic information, and then construct the semantic link between MIRs based on the semantic description of MIRs. Moreover, we should adopt a hierarchical model to manage these semantic links.

In section 2, we present a virtual hierarchical semantic framework for MIG, including the MIR semantic description model and organization model. On basis of this framework, we present a virtual semantic resource routing algorithm for rapidly and precisely searching MIRs in section 3. In section 4, we analyze the performance of VSRR. In section 5, we conclude the paper and point out the future work.

2 The Virtual Hierarchical Semantic Framework for MIG

2.1 The Description of the Framework

Considering the semantic relativity between MIRs and the efficiency of sharing MIRs, we propose the virtual hierarchical semantic framework which is the virtual hierarchical structure constructed by grid node, virtual semantic resource router and virtual semantic link. Virtual semantic resource routers are organized as a net, which divides the MIG into a lot of virtual semantic resource sub-grids (VSRSs) to reduce the scale of the RSCs by truncating the semantic chains. The MIRs in a VSRS are linked in semantic by RSC, and the MIRs in diverse VSRS are linked in semantic by RSC and virtual semantic resource routers. Therefore, if only getting one requested MIR, we can obtain all relative requested resources via RSCs and virtual semantic resource router. Figure 1 illustrates the virtual hierarchical semantic framework.

Definition 2.1: *The multimedia information grid* is defined as $G = (N, R, \delta, SC, VR)$, where N denotes a set of the nodes in the MIG; R is a set of the MIRs; $\delta : R \rightarrow N$, denotes the distribution of MIRs in nodes; SC represents a set of RSCs between MIG nodes; VR denotes a set of the virtual semantic resource routers in MIG.

Virtual semantic resource router controls the registration, update, correction, deletion of the RSCs and MIRs. MIR semantic chain table and semantic resource

Definition 2.5: *The multimedia information resource* is defined as $r = (r_{syn}, r_{sem}, r_p, r_a)$, where r_{syn} denotes the syntax description of MIR; r_{sem} denotes the semantic description of MIR; r_p is the permanence of the MIR; r_a is the availability of the MIR, and it includes the reliability and the employing cost.

Definition 2.6: *The semantic description for MIR* is defined as $r_{sem} = (S_H, S_M, S_L)$, where S_H denotes the high level semantic description for MIR, including the name, type and profile description of MIR; S_M denotes the scenario and effect of the presentation of the MIR in middle level semantic; S_L represents the detail description in low level semantic. S_H, S_M, S_L can be represented as $((s_1, s_2, \dots, s_n), (w_1, w_2, \dots, w_m), \mu)$, where $s_i (1 \leq i \leq n)$ is the semantic index words which can simply and precisely describe the semantic of the multimedia resources; $w_i (1 \leq i \leq m)$ is the weight value of every index words; μ is the abstract degree of every level semantic description.

2.3 The Organization Model

The MIRs are organized by RSCs. RSCs are built according to the syntax and semantic relativity between original MIR and destination MIR. By this, we can rapidly and precisely search several aggregates of the requested MIR assigned diverse relevant degree. Every grid node has a MIR description table involving the description objects of every available MIR. Semantic chain includes three level semantic chains belonging to diverse semantic level. While matching in every level, every index word in this semantic level of source MIR is separately matched with every index word in relevant semantic level of destination MIR in many relevant ontology regions. Figure 2 illustrates the structure of the RSC.

Definition 2.7: *The resource semantic chain* is defined as $sc = (N_H, N_{sc})$, where N_H represents the head node; N_{sc} is a set of nodes, every node is formed as (r, T_{link}) , where r denotes the source MIR; T_{link} denotes the table of bidirectional links, and it is a list of items formed as $(ID_{ndm}, d_{ndm}, ID_{pdm}, d_{pdm})$, where ID_{ndm} is the pointer of next destination MIR; d_{ndm} is the matching degree between source MIR and next destination MIR. ID_{pdm} is the pointer of previous destination MIR; d_{pdm} is the matching degree between source MIR and previous destination MIR. d_{ndm} and d_{pdm} can be formed as $(d_H, d_M, d_L, w_H, w_M, w_L)$, where, d_H, d_M, d_L represent the high level semantic matching degree, the middle level semantic matching degree and the low level semantic matching degree, respectively; w_H, w_M, w_L represent the weight value of three semantic level, respectively.

Definition 2.8: *The resource description table T_{mr}* is defined as a list of items formed as (r, sc) , where r is MIR; sc is the resource semantic chain updated every fixed interval.

Definition 2.9: *The matching degree* is defined as:

$$d = \xi_r(r_s, r_d) = \sum_{j \in (H, M, L)} \mu_j \sum_{s_{oij} \in S_{S_j}, s_{dij} \in S_{D_j}} w_{ij} \xi_s(s_{oij}, s_{dij}),$$

where $\xi_r : R \times R \rightarrow (0, 1)$ and $\xi_s : r_{sem} \times r_{sem} \rightarrow (0, 1)$ are matching functions; r_s is the source MIR; r_d is the destination MIR; s_{oij}, s_{dij} represent the high, middle and low level semantic index words of the source MIR and destination MIR, respectively; w_{ij} represents the weight of every index word in high, middle and low level semantic descriptions, respectively; μ_j represents the weight of high, middle and low semantic descriptions, respectively; S_{Sj}, S_{Dj} represent the high level, middle level and low level semantic descriptions of source MIR and destination MIR, respectively.

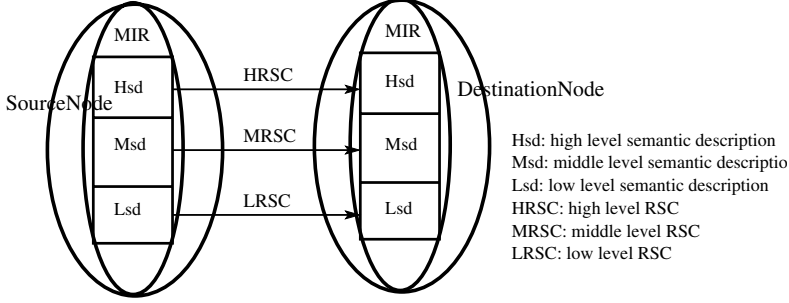


Fig. 2. Multilayer Resource Semantic Chain

We built RSC for every description object in the MIR description table according to semantic matching result, and assign the RSC a matching degree in terms of semantic matching result. The length of every RSC and the quantity of RSCs are limited by network distance and semantic matching degree. Every grid node registers its all RSC to virtual semantic resource router. While the MIR is altered in grid node, this grid node must update relative RSCs and inform the virtual semantic resource router.

Algorithm 2.10: *Constructing RSC*

This algorithm constructs all RSCs in a VSRS, and registers these RSCs to virtual semantic resource router.

Input: matching degree threshold value d_T

Output: T_{mr} in source node and the T_s in virtual semantic resource router.

Procedure: *constRSC*(d_T)

- (1). Build the vocabulary lists for all ontology domains T_s in virtual semantic resource router. Every node downloads the requested T_s from virtual semantic resource router;
- (2). Chose source r from T_{mr} in source node, match it with other r in T_{mr} in turn; if $d \geq d_T$, then fill the destination r in the sc of the source r ;
- (3). Match the source r with the rs in other nodes in this VSRS, matching method is similar with step (2);
- (4). Repeat the step (2,3) until this r is matched with all other rs in this VSRS;

- (5). Chose other rs in source node in turn, and repeat step (2,3,4) until all rs in this node are matched;
 - (6). Chose other node as source node, and repeat step (2,3,4,5) until all rs in VSRS are matched. Then, every sc and relative r construct a graph, and adopt minimal spanning tree algorithm for pruning the redundant semantic links, and regard the root r as the representation of the sc ;
 - (7). Fill all scs in the T_{mr} in source node and the T_s in virtual semantic resource router;
- End *constRSC*.

3 Virtual Semantic Resource Routing Algorithm

In searching MIR, we divide the searching request into three classes: the minimal time searching, the optimum searching and the aggregate searching. The minimal time searching seeks the first MIR meeting the requirement; the optimum searching is finding the optimum MIR meeting the requirement; In the aggregate searching, the searching result is a multimedia resource aggregate involving all multimedia resources over matching bounded value and its matching degree.

In MIG, when searching some MIRs, we frequently need the aggregate of optimum MIRs. So, in this algorithm, the searching request from one grid node firstly reaches the MIRs via relative RSCs according to matching degree, and synchronously, the request is delivered to virtual semantic resource router which forwards the request to the adjacent routers including the requested MIR based on the high level RSCs in semantic resource routing list. Then, routers having received the request forward the request to their adjacent routers and search the aggregate of requested resource in their VSRS by the RSCs in MIR semantic chain table.

Definition 3.1: *The matching aggregate of MIRs* is defined as R_m which is a set of elements represented as (r, d_H, d_M, d_L) , where r denotes the MIR; d_H, d_M, d_L represent the matching degrees of r with source MIR in high, middle and low semantic level, respectively.

Definition 3.2: *The requested MIR* is defined as $r_r = (r, d_T, r_T)$, where r is the MIR; d_T is the matching degree threshold; r_T is the availability threshold.

Algorithm 3.3: *The aggregate searching of MIR*

Input: The requested MIR r_r

Output: The matching aggregate of MIRs R_m .

Procedure: *searching*(r_r)

- (1). Choose r from its T_{mr} , if $(r.d \geq d_T) \cap (r.r_a \geq r_T)$ then go to step (2) else go to step (4); Simultaneously, source node sends request message including r_r to virtual semantic resource router of this VSRS;
- (2). Fill this r in the R_m , and get the sc of this r ;
- (3). Get the head r in sc , if treat it at first time, then find rs along with sc in T_{mr} by breadth-first traversal algorithm, and repeat if $(r.d \geq d_T) \cap (r.r_a \geq r_T)$ then fill this r in the R_m , until completely search the every MIR in sc 's tree; then go to step (5);

- (4). Choose the next r in T_{mr} and go to step (1) until all rs in T_{mr} are visited;
 - (5). The virtual semantic resource router looks up its T_c , if $\exists sc, (sc.d \geq d_T)$ then go to step (3);
 - (6). The virtual semantic resource router looks up its T_r , get the result aggregate of a set of virtual semantic resource routers which meet the condition $(sc.d \geq d_T)$, and then this virtual semantic resource router sends the request message including r_T to the routers in aggregate;
 - (7). For every router having received the request, repeats steps (5,6) and get the result aggregate R_m to source Grid node. Finally, the source node synthesizes all R_m ;
- End *searching* .

4 Performance Analysis

We consider the performance of VSRR algorithm of the searching aggregate of MIRs. Because this algorithm basically involves two sections, one is the constructing procedure of the RSC and the T_r , the other is the searching aggregate procedure of MIRs. The semantic resource router table has accomplished initially, and so we can ignore the cost of this part when the MIRs are stable. So the complexity of this algorithm mainly attributes to the searching aggregate procedure of MIRs .

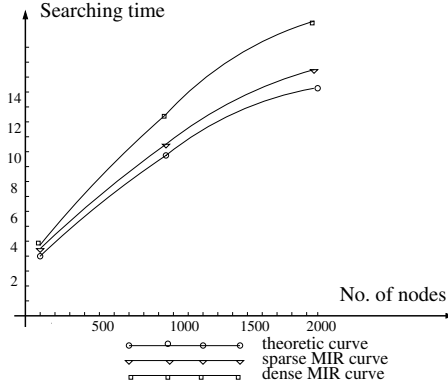


Fig. 3. The time complexity of this algorithm

In the searching aggregate procedure of MIRs, if ignoring the time of sending back the result and composing the result, the time of searching is mainly defined by the longest searching path. The cost of the longest searching path is $C_L = length \times C_X + C_F$, where, *length* denotes the number of virtual semantic resource router; C_X represents the cost of looking up the semantic resource router table; C_F represents the cost of looking up the MIR semantic chain table. So the time complexity of this algorithm is: $O(V_S) = O(length \times C_X + C_F)$.

So we can conclude the time complexity graph of our algorithm. The algorithm reduces the improving velocity of time complexity accomplishing with the improvement of the scale of the grid and the number of the MIRs. We simulate the VSRR algorithm in different scales of MIG: small-scale MIG(5 VSRs, 100 nodes), middle-scale MIG(10 VSRs, 1000 nodes) and large-scale MIG(16 VSRs, 2000 nodes), and in diverse MIR distribution (dense, sparse). We consider the time of looking up the table T_r as a time unit. The results are shown as Figure 3.

5 Conclusions and Future Work

In this paper, we propose the virtual semantic resource routing algorithm to improve the efficiency of searching multimedia information resource. For implementing this algorithm, we construct the virtual hierarchical semantic framework for multimedia information grid. In this framework, we study the description model for the MIR and the description and construction of the RSC between MIRs. Finally, we describe aggregate searching algorithm in VSRR and then analyze the performance of VSRR. But there are still many problems, for example, exploring more efficient searching algorithm for different MIR distribution, more precisely evaluating the cost of MIR searching and simulating the algorithm in large-scale MIG.

References

1. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.
2. I. Foster, C. Kesselman, and S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, *Int'l J. Supercomputer Applications*, 2001, 15(3), pp. 200-222.
3. Fox, G., *Integrating computing and information on Grids*, *IEEE Computational Science and Engineering*, 2003, 5(4), pp. 94 - 96.
4. Z.Xu, X.Li, and G.You, *Architectural study of the Vega Information Grid*, *Journal of Computer Research and Development*, 2002, 39(8), pp. 948-951.
5. Cannataro, M., Talia, D., *Semantics and Knowledge Grids: Building the Next-generation Grid*, *IEEE Intelligent Systems*, 2004, 19(1), pp. 56-63.
6. Johnston, W.E., *Semantic services for grid-based, large-scale science*, *IEEE Intelligent Systems*, 2004, 19(1), pp. 34 - 39.
7. A.B.Benitez, et al., *Semantics of multimedia in MPEG-7*. 2002 International Conference on Image Processing, 2002, 1(1), pp. 137-140.
8. M.Abdel-Mottaleb, et al., *Multimedia descriptions based on MPEG-7: extraction and applications*. *IEEE Transactions on Multimedia*, 2004, 6(3), pp. 459-468.
9. A.Iamnitchi, I. Foster, and D.Nurmi. *A peer-to-peer approach to resource discovery in grid environments*. Technical Report TR-2002-06, University of Chicago, 2002.
10. W.Li, Z.Xu, F.Dong, and J.Zhang. *Grid resource discovery based on a routing-transferring model*. *The 3th International Workshop on Grid Computing*, pp. 145-156, Baltimore, MD, 2002.

Digital Library Application Grid – An Opportunity to Open Cultural Infrastructure*

Zhongzhi Luan¹, Depei Qian¹, Xiaoshe Dong², Xingjun Zhang², and Yunchun Li¹

¹ School of Computer Science and Engineering, Beihang University, Beijing, 100083
rick710055@263.net

² Dept. of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, 710049
xsdong@mail.xjtu.edu.cn

Abstract. The cultural infrastructure provides for the transmission of culture from creators to audiences. The access to this infrastructure is the key issue. The network technology will radically transform the interaction with knowledge. The base infrastructure will be knowledge networks rather than data networks. Digital libraries would become the universal knowledge repositories and communication conduits of the future. These digital libraries should enable any citizen to access human knowledge any time and anywhere, in a friendly, multi-modal, efficient and effective way. Ideally the infrastructure combines concepts and techniques from Grid computing. It would be an opportunity to open cultural infrastructure.

1 Introduction

The cultural infrastructure is a complex system of relationships among individuals and public, private, for-profit and not-for-profit institutions. This system provides for the transmission of culture from creators to audiences. There are literally millions of access points into the cultural infrastructure through museums, libraries, universities, historical societies, web sites, broadcasts, streaming video, magazines and live performances. There is no shortage of cultural expression, but for many people, getting at that culture can be a real challenge [1].

People produce, receive, and exchange cultural experiences with one another in many different ways and forms. According to [1], there are three primary means of access: physical, traditional media and new media. New media include the information technologies that are quickly becoming pervasive throughout society – the World Wide Web, video and audio streaming, online searchable archives, and broadband connectivity.

The network technology will radically transform the interaction with knowledge. Traditionally, online information has been dominated by data centers with large collections indexed by trained professionals. The rise of the Web and the information infrastructure of distributed personal computing have rapidly developed the technolo-

* This work was supported by the National Science Foundation of China (NSFC) under the grant No.90104022, and the National Key Basic Research Program of China (973 Project) under the grant No. G1999032710.

gies of collections for independent communities. In the future, online information will be dominated by small collections maintained and indexed by the community members themselves. The information infrastructure must accordingly be radically different to support indexing of community collections and searching across such small collections. The base infrastructure will be knowledge networks rather than data networks [2].

In the past several years, a large number of Digital Library systems have been developed. Each system is typically built from scratch and develops its own techniques, focusing on a specific type of information or services that it supports and addressing the needs of a specific application or domain. After all this experience, it has become clear that the future of digital libraries goes well beyond what these past efforts may indicate individually. Furthermore, it is evident that traditional management of plain text makes way for that of enriched documents with embedded knowledge.

We see the potential for digital libraries to become the universal knowledge repositories and communication conduits of the future, a common vehicle by which everyone will access, discuss, evaluate, and enhance information of all forms.

2 Digital Library Technology

The emerging field of digital libraries brings together participants from many existing areas of research. From a database or information retrieval perspective, digital libraries may be seen as a form of federated databases. From a hypertext perspective the field of digital libraries could seem like a particular application of hypertext technology. From a wide-area information service perspective, digital libraries could appear to be one use of the World Wide Web. From a library science perspective, digital libraries might be seen as continuing a trend toward library automation. There is some truth to these perspectives (as well as others) but none address the field as a whole and its research agenda. Digital library research must both respect the existing tradition of our physical libraries and transcend current practice in developing a new, broader research agenda [3].

An element of a library is a constituent part of the library. It is helpful to consider three broad classes of library elements: data, metadata, and processes. Data are library materials. Metadata are information about the library and its materials. Processes are active functions performed over library elements. A domain of the library is the universe from which the library materials are drawn. A physical library deals primarily with physical data, whereas a digital library deals primarily with digital data. Of course most modern libraries deal with both, but it is useful for sake of discussion to consider hypothetical "all-physical" and "all-digital" libraries as foils.

The field of digital libraries presents a set of complex issues, and solutions to these problems will require a blending of approaches from a variety of fields. Claims that any one technology has solved all of the issues posed in the design and implementation of digital libraries fail to address the entire problem. Instead, any successful attempt at constructing a digital library system will need to address issues raised by considering the many different kinds of digital library elements throughout the various levels of the general digital library system architecture.

Looking forward, the future evolution of the digital library field can be viewed from various perspectives. We see key dimensions corresponding to these perspectives, along which advancement of the field can be evaluated [4].

From architecture dimension, we see increasing capabilities and dynamicity as more sophisticated system and network architectures develop. From early standalone system's digital library, to homogeneous distributed systems and to heterogeneous distributed systems and in the future dynamic virtual digital libraries.

From interoperation dimension we see an increasing number of aspects in which DLs can interoperate, such as search and retrieval, repository, security and authorization, quality assessment and subscription.

From information dimension, the sophistication with which individual DLs reason about the information they hold and are able to communicate with other DLs is delineated. Some points on this dimension are data, Meta-data, extensibly structured information and knowledge representation.

From service dimension, the complexity of processing that DLs and federations of DLs can manage on behalf of clients is characterized, such as Web service, workflow management and agent hosting.

In each dimension, deployed systems tend to be at the first or second point, while research systems are further along (though not necessarily along all dimensions).

3 Using Grid Technology on DL

Future digital libraries (DLs) should enable any citizen to access human knowledge any time and anywhere, in a friendly, multi-modal, efficient and effective way. Ideally the infrastructure combines concepts and techniques from the following fields: Peer-to-Peer data management, Grid computing and service-oriented architecture as the next generation digital library technologies.

Peer-to-peer (P2P) architectures allow for loosely coupled integration of information services and sharing of information such as recommendations and annotations. Different aspects of peer-to-peer systems (e.g. indexes, and P2P application platforms) must be combined. Grid computing is needed because certain services within digital libraries are complex and computationally intensive (e.g., extraction of features in multimedia documents to support content-based similarity search or for information mining in bio-medical data). The service-oriented architecture (SoA) provides mechanisms to describe the semantics and usage of information services. Moreover, in a SoA we have mechanisms to combine services into workflow processes for sophisticated search and maintenance of dependencies.

It seems that elements of all these directions should be combined in a synthesis for future DLs architectures. And the adoption of Grid technology would solve many issues of digital library because of its resources sharing and operation coordinate features. We are attempting to build an infrastructure based on Grid computing with National Library. The key issues we plan to solve are organization and management of distributed resources, tasks deployment and resources coordination, distributed management and query of huge number of heterogeneous data, organization and coordination of dynamic services, reconstruction approach of system and distributed presentation, storage, transmission and retrieval of metadata. We think it is a good

opportunity on building an open culture infrastructure. We hope our work could do some contribution to this goal.

4 Summary

The cultural infrastructure provides for the transmission of culture from creators to audiences. The access to this infrastructure is the key issue. The network technology will radically transform the interaction with knowledge. The base infrastructure will be knowledge networks rather than data networks. Digital libraries would become the universal knowledge repositories and communication conduits of the future. These digital libraries should enable any citizen to access human knowledge any time and anywhere, in a friendly, multi-modal, efficient and effective way. Ideally the infrastructure combines concepts and techniques from Grid computing. We are attempting to build an infrastructure based on Grid computing. It would be an opportunity to open cultural infrastructure.

References

1. Allison, Brugg, Bawden, Access and the Cultural Infrastructure, *Art, Culture & the National Agenda Issue Paper*, Nov.2002.
2. Schatz, B. R. (1997) Information Retrieval in Digital Libraries: Bringing Search to the Net. *Science*, 275(5298):327-334, January.
3. Peter J. Nürnberg, et. al, Digital Libraries: Issues and Architectures, Digital Libraries '95 Proceedings.
4. DELOS-NSF Working Group, Digital Library Information-Technology Infrastructures, 2003.

A JDO Storage Cluster Based on Object Devices*

Youhui Zhang, Dongsheng Wang, Chongnan Gao, and Weimin Zheng

Institute of High Performance Computing Technology
Dept. of Computer Science, Tsinghua Univ.
100084, Beijing, P.R. China
zyh02@tsinghua.edu.cn

Abstract. Today's Internet services demand the storage platform to possess high performance and simple programming interfaces. This paper presents the design and implementation of a distributed objects storage cluster system, which employs Network-Attached Object Storage Device as the low-level storage device to support structured-data directly to eliminate the problem of conventional storage systems. This system provides a unified, transparent and object-oriented view of the storage devices of the whole cluster and greatly simplifies distributed service development. Based on this system, an open source mail server is enhanced to be a distributed one while only a few source files are modified. Testing shows that the performance of this enhanced mail system can achieve 2/3 of the ideal upper limit, which is evidently higher than the original one based on file systems.

1 Introduction

For the ever-increasing storage demands of Internet services, it is argued that RDBMS (and also parallel RDBMS) often introduces too much overloads because of the strict ACID semantics and lacks scalability. On the other hand, file systems (and also distributed file systems) are often considered too general in the sense that the system knows nothing about the structure of user data, thus leaving all the work of organizing, accessing and querying data to application developers. In addition, using block-based disks as storage devices will cause a data-model-mismatch problem between applications and the storage systems, because disks only provide a simple block-access interface. Therefore, one or more "server" computers have to copy and convert data between the storage (peripheral) network and the client (local area) network. It is called distributed server bottlenecks [1], which impairs the system scalability.

Fortunately, Object-based Storage Device (OSD) [2] can be employed to mitigate these problems. As we know, OSD will provide more complicated access interfaces than traditional disks and support variable-length data object, which will introduce two main advantages.

- The data-model-mismatch problem is mitigated.
- Object-based storage systems separate the data and metadata management [3].

* Supported by High Technology and Development Program of China (No. 2002AA1Z2103).

OSDs manage low-level data storage tasks such as request scheduling and data layout, and present a object-based data access interface to the rest of the system. Metadata servers are not involved in the storage and retrieval of data, allowing for very efficient concurrent data transfers between large numbers of clients and OSDs, which improves the system performance and scalability.

Enlightened by this idea, we design a new network attached object-based storage device (NAOSD). Compared with the previous devices, NAOSD owns the following extra features.

- Structured-data storage is supported directly while the other OSDs only support variable length data with some attributes.
- Simple Computing abilities: Query and sorting are both supported by NAOSD.

Based on the cluster of NAOSDs, we design and implement a distributed JDO (Java Data Objects) storage system, which has the following functions.

- It supports transparent persistence in the Java programming language. User services written in Java can transparently access objects stored in the system. The client interface is compatible with Java Data Objects API (version 1.0)[4].
- It is an object-oriented data management layer as a cluster infrastructure software with transaction support, specifically for the construction of Internet services.
- Peer-Client and modular design are adapted. Many storage properties, including the storage capacity, network bandwidth and throughput, are highly scalable.

Owing to the JDO storage system, development of some cluster services will be simplified because the storage layer has solved all distributed storage issues and developers can focus their mind just on service functions. It spent us less of one week to port Apache James 2.1.3 [5] mail server for the distributed JDO system. That is, only a few modifications are needed to enhance the mail server to be a distributed one.

The rest of this paper is organized as follows. Section 2 describes some related research work. NAOSD is introduced in Section 3 briefly. Section 4 presents the architecture and the implementation of its prototype. The mail system is described in Section 5. The next section introduces the performance testing and analyzes the results. Section 7 summarizes the whole paper.

2 Related Works

Most current clustered services use home developed service-specific solutions to achieve storage. For example, the Porcupine clustered email system [6] uses its own distributed storage manager and doing replication of critical data all by its own. Although it is an efficient and scalable service, this approach mixes application logic with low-level data management details.

Ninja [7] project implements DDS (distributed data structure) that presents a conventional single-site data structure interface to service authors, but partitions and replicates data across a cluster. Now a distributed hash table DDS is implemented.

OceanStore [8] is a global persistent data store designed to scale to billions of users. It provides a consistent, highly available, and durable storage utility atop an infrastructure comprised of servers. Any computer can join the infrastructure, contributing storage or providing local user access in exchange for economic compensation. Users

need only subscribe to a single OceanStore service provider, although they may consume storage and bandwidth from many different providers.

On the other hand, some research projects, including NASD [9], Attribute based Storage [10], Active Disks [11], bring forward the idea of Object-based Storage Device (OSD), which means that some overloads owned by traditional file servers are offloaded to peripheral storage devices. However, the data object of these projects likes the inode of file systems and the main targets are to simply the design of file systems. Distributed object-based file systems, currently used in systems such as Lustre [12] and OBFS [13], are built on OSDs. They abstract away file storage details such as allocation and scheduling, semi-independently managing all of the data storage issues and leaving all of the file metadata management to the file manager.

Our goal is to design a cluster storage system with high performance, reusable management layers and simple programming interfaces to solve most storage issues, which let developers focus their mind on the service logic. So, NAOSD is designed with the extra features described in Section 1 and the prototype of a distributed object storage system based on NAOSD is implemented.

3 NAOSD

3.1 Features

NAOSD is designed to provide upper-level services with more abstract and higher access interfaces. It owns the extra features, including object-like access interface, simple computing abilities and storage self-management.

Three advantages are introduced when using NAOSD as the storage device.

- Direct storage-device-to-client transfers are supported to eliminate the traditional server bottleneck.
- Some server functions are ported to the storage devices to leverage the parallelism available in systems with large numbers of disks.
- The amount of data on the interconnection is reduced. It is especially useful for those data-intensive Internet services.

This paper does not focus on the detailed performance analysis of NAOSD and its implementation. A model for determining the potential performance benefits of a service in a system using NAOSD is presented in [14].

3.2 Access Interfaces

In current prototype NAOSD is simulated on Berkeley DB [15], an embedded database. Now it cognizes some basic data types, including integer, char, string, date and so on, which can combine to the structured object it supports. An Object is identified a 128bit integer, OID. In addition, the following basic access interfaces are provided.

- Register: To open a session with the NAOSD and get some device information.
- InsertObject: To insert an object with OID.
- FetchObject: To read an object, which is identified by the provided OID.
- DeleteObject: To delete an object, which is identified by the provided OID.

- **GetExtent:** To get all objects whose object type is as same as the provided type.
- **QueryExtent:** To get all objects whose object type is as same as the provided type and matches the provided filter condition(s). For example, one basic filter will look like “Object Type= xxxxx && the Value of one Object field =yyyyy”.

All of the commands and results are transferred through the network between the NASOD and the client.

4 Overview of the JDO Storage Cluster

4.1 The Architecture

The storage system is defined as a self-contained object oriented data management layer running on a cluster to handle storage requests of Internet services on the same cluster.

The Internet services connect to storage components known as Peer Servers to access data. They are called Peer Servers because they are inherently identical to each other from users' view, each presenting a single image of the whole system and they communicate with each other in a peer-to-peer style.

Brick is the instance of NAOSD that provides storage and query interfaces for structured-data employing the power of embedded processors.

Within the service process, a library named TODSLib maps user API calls to messages sent to Peer Servers and parse results from them. Currently a Java version of TODSLib is implemented. As mentioned before, it implements transparent persistence and is compliant with the Java Data Objects API.

The Meta-Server maintains system configuration and meta-data. It is replicated and thus assumed fault-tolerant, providing a safe place for critical global information. System configuration includes location (IP, port) and parameters of all components such as Peer Servers and Bricks. This ensures centralized management of the whole system.

4.2 The Prototype

The prototype of the distributed JDO storage system is implemented and its hardware platform contains a cluster of PC servers connected with 100M Ethernet. The whole system is coded with JDK 1.4 other than Bricks that are implemented in C language based on Berkeley DB. With advanced features such as XA transaction support and replication, and a long evolving history, Berkeley DB provides a very stable foundation for our work.

5 The Mail Server

We ported Apache James mail system for our distributed JDO system, which is built on top of the Avalon Framework and contains several service components as follows.

- POP3 Service

It provides full compliance with the specification and maximum compatibility with common POP3 clients to retrieve email messages.

- SMTP Service

SMTP (Simple Mail Transport Protocol) is the standard method of sending and delivering email on the Internet. The mail server provides a full-function implementation of the SMTP specification.

- NNTP Service

NNTP is used by clients to store messages on and retrieve messages from news servers. The system provides the server side of this interaction by implementing the NNTP specification.

Some other services, such as FetchPOP, SpoolManager, Matcher, and Maillet are implemented, too. All of these source files are not modified in our implementation except of Repository components. A number of different repositories to both store message data (email, news messages) and user information are used in the mail system.

Aside from what type of data they store, repositories are distinguished by where they store data. There are four types of storage - File, Database, DBFile and the distributed JDO storage system. We introduce the last type.

In the mail system, all repositories are instances of the MailRepository interface. So it is necessary for us to implement methods of MailRepository interface based on the distributed JDO storage system, which are listed as follows.

- Iterator **list()** List string keys of messages in repository.
- boolean **lock**(String key) Obtains a lock on a message identified by key.
- void **remove**(MailImpl mail) Removes a specified message
- void **remove**(String key) Removes a message identified by key.
- MailImpl **retrieve** (String key) Retrieves a message given a key.
- void **Store**(MailImpl mc) Stores a message in this repository.
- boolean **unlock**(String key) Releases a lock on a message identified the key.

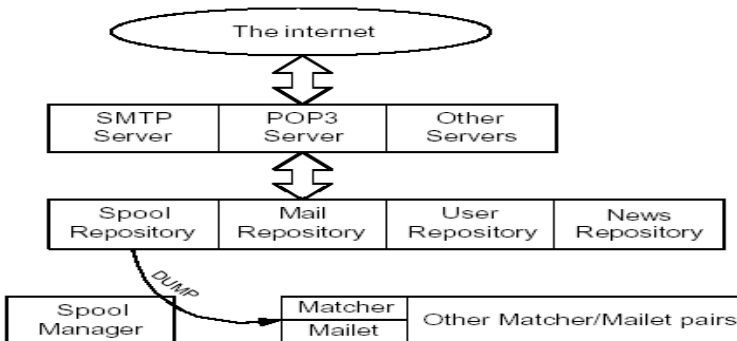


Fig. 1. The architecture of mail server

Correspondingly, TODSMailRepository class is defined, which also implemented these interfaces. Most features provided by JDO, including transaction updating, ob-

ject query, class extents and transparent persistency are all employed in our code. Similarly, a new UserRepository class is implemented to save the account info. Owning to JDO features, object persistency and is easy to implement and some example codes are presented as follows.

```
try {

    Transaction tx = pm.currentTransaction();

    tx.begin();

    pm.makePersistent(todsmail);

    tx.commit();

} catch(Exception e) {

    e.printStackTrace(System.err);

}

}
```

6 Performances

Performance experiment results are presented in this section. Our test environment is a server cluster and each node is equipped with 2 Intel Pentium III Xeon processors at 750 MHz, 1 GB of RAM and a 36 GB 10000 RPM SCSI disk. The network is 100M fast Ethernet. All nodes run Red hat Linux 7.2. The system and test programs were run with Sun JDK 1.4.0 for x86 Linux. One node is selected as the Peer Server running mail service and several others selected as Bricks.

For mail system the most important performance is the throughput, that is, how many messages the system can deal in one time unit. A simulated running environment is constructed where several simultaneous threads send SMTP requests to our mail server continuously. The size of every email is set within a predefined range and the amount of requests is adjustable. In our testing, the maximal email size is 10k bytes and 1,000,000 accounts are created.

To evaluate the ideal upper limit of performance, the mail system is modified that any received email will be discarded without storage. So the throughput is only restricted by the network bandwidth and the processing power of mail server. Under these conditions, testing shows that the ideal upper limit of one server is about 2900 messages per minute.

Next, the local file system is employed as MailRepository and the throughput is about 1100 messages per minute.

In contrast, same testing is executed on top of our TODSMailRepository and results are listed in Table 1. From the results, one conclusion can be drawn that the system bottleneck is the storage layer. Our system with four Bricks achieves 69% of the ideal upper limit, which is evidently higher than the one based on file systems. But the throughput cannot increase with the number of bricks continuously. We think that the Peer Server becomes the bottleneck when the number of bricks exceeds 4.

Table 1. Testing results

Condition	Msg/Minute
The upper limit	2900
Based on file systems	1100
Based on the JDO system (one peer server and one Brick)	1500
Based on the JDO system (one peer server and two Brick)	1800
Based on the JDO system (one peer server and three Brick)	1900
Based on the JDO system (one peer server and four Brick)	2000
Based on the JDO system (one peer server and five Brick)	2000

7 Conclusion

This paper presents the design and implementation of a distributed Java Data Objects storage system on a cluster of NAOSDs, which provides a unified, transparent and object-oriented view of the storage devices of the whole cluster and greatly simplifies distributed service development. Based on this system, Apache James 2.1.3 mail server is enhanced as a distributed one and testing results show that its performance is evidently higher than the original one that employs the local file system as the storage repository.

Compared with RDBMS and file systems, the impedance mismatch problem is avoided and most distributed storage issues are solved in the JDO storage cluster, which include object query, distributed transaction and class extents. Then developers can focus their mind on the service logic.

References

1. Garth A. Gibson, David F. Nagle, William Courtright II, etc. NASD Scalable Storage Systems. In the Proceedings of USENIX 1999, Linux Workshop, Monterey, CA (1999).
2. R. O. Weber. Information technology – SCSI object-based storage device commands (OSD). Technical Council Proposal Document T10/1355-D, Technical Committee T10, (2002).
3. Garth A. Gibson, Brent B. Welch, David F. Nagel, and Bruce C. Moxon. Object Storage: Scalable Bandwidth for HPC Clusters. In proceedings of Cluster World Conference and Expo (2003).
4. C. Russell, Java Data Objects 1.0 Proposed Final Draft, JSR12, Sun Microsystems Inc., available from <http://access1.sun.com/jdo> (2001).
5. Apache James website: <http://james.apache.org/>.
6. Yasushi Saito and Brian N. Bershad and Henry M. Levy. Manageability, Availability and Performance in Porcupine: A Highly Scalable, Cluster-based Mail Service, Symposium on Operating Systems Principles (1999).
7. Steven D. Gribble and Eric A. Brewer and Joseph M. Hellerstein and David Culler. Scalable, Distributed Data Structures for Internet Service Construction, 4th Symposium on Operating System Design & Implementation, San Diego (2000).
8. John Kubiawicz, David Bindel, Yan Chen, etc. OceanStore: Architecture for Global-Scale Persistent Storage. in Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000).

9. G.. A. Gibson, et al., A Case for Network Attached Secure Disks, Tech. Report CMU-CS-96-142, Carnegie Mellon University (1996).
10. Elizabeth Shriver, A Formalization of the Attribute Mapping Problem. HP Labs Technical Reports, HPL-1999-127.
11. Erik Riedel, Christos Faloutsos, Garth A. Gibson, etc. Active Disks for Large-Scale Data Processing. IEEE Computer, Vol.34, No.6 (2001). 68-74.
12. P. J. Braam. The Lustre storage architecture, <http://www.lustre.org/docs/lustre.pdf> (2002).
13. Feng Wang, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long. OBFS: A File System for Object-based Storage Devices. In the 21st IEEE / 12th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST2004), College Park, MD (2004).
14. Youhui Zhang, Weimin Zheng. Research on a new type of network-attached object based storage device. Tien Tzu Hsueh Pao/Acta Electronica Sinica. v.31,n.5. (2003).
15. Sleepycat Software Inc., Berkeley DB Programmer's Tutorial and Reference Guide, available at <http://www.sleepycat.com> (2001).

iNASC: A iSCSI-Based NAS Storage Cluster

Dezhi Han¹, Changsheng Xie², and Faling Yi²

¹ department of computer science, JiNan University, Guangzhou
Guangdong, P.R. China, 510632
dezhihan88@sina.com

² Huazhong University of Science and Technology, Wuhan
Hubei, P.R. China, 430074

Abstract. To work out a network storage system with a large capacity, a high I/O speed, availability and scalability, a iSCSI-based NAS cluster (iNASC) has been designed. Firstly, the iNASC integrates multi-NAS with the storage virtualization and redundancy technologies, which can provide a greater capacity, higher availability and scalability; secondly, the iNASC simultaneously serves for both the file I/O and the block I/O with an iSCSI module, which has the advantages of NAS and SAN; thirdly, the parallel file I/O performance is improved by a parallel FTP module; finally, the iNASC can provide higher I/O speed by a high speed IP channel which implements the direct data transfer between the NAS and client. In the experiments, the iNASC has ultra-high-throughput for the file I/O and block I/O requests.

1 Introduction

With digital information on networks increasing at a tremendous speed, it is urgent to work out a network storage system with a large capacity, a high I/O speed, availability and scalability. At present, more and more efforts have been made to improve the traditional NAS. The NAS has such advantages as sharing files in a hetero-architecture, easy installation, good connection compatibility and network adaptation, low costs and so on [1], [2]. In practice, however, the NAS also exists some disadvantages,: (1) It does not support the block I/O protocol except the NFS and the CIFS; (2) In the aspects of system resources integration and management, it can only integrate the disk resources in a single NAS device and cannot span different NAS device; (3) When the data is copy between NAS devices, it occupies most of the LAN bandwidth, which will hamper users' access. To solve the above-mentioned problems, the paper introduces an iNASC which is made up of the multi NAS.

2 The iNASC Architecture

At present, the NAT (Net Address Transfer) has been used to construct the NAS Cluster, in which the whole NAS Cluster has only one metadata server to receive all

kinds of requests. The users' requests are dealt with by the metadata server at first, then they are passed to the corresponding NAS storage server; afterward, the requested data are transferred to the metadata server by an internal network and finally to users by an external network. This model can support any OS and any private network with only one external IP address. But all I/O requests and data must be processed and transferred by the metadata server, which leads to a system bottleneck. To solve this problem, a new model of the metadata server is put forward in this paper, the metadata server only receives users' requests, provides a single storage view to the users, equilibrates the loads and checks users identities. The data is directly transferred between the NAS and the clients via the high speed IP channel. The read/write process is shown in Figure 1. By this way, the scheduling and processing capabilities of the metadata server are significantly improved.

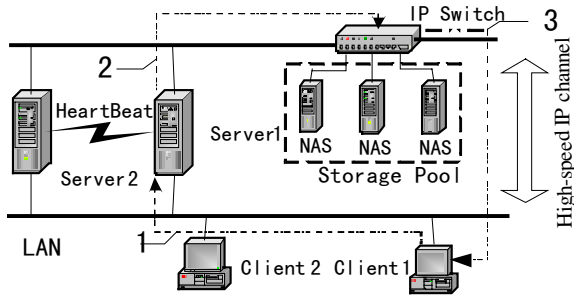


Fig. 1. The iNASC architecture

3 System Design

This paper proposes three mechanisms to solve the problems of the common NAS cluster. The first one provides single system image by storage virtualization technology; the second provides the block I/O service and the file I/O service simultaneously with iSCSI technology; the third provides a server channel and a high network-attached channel to clients simultaneously; the fourth processes multi-users, multi-tasks requests in parallel.

3.1 Design of a Uniform I/O Space File System

In this paper we design and implement a uniform I/O space file system between the VFS layer and the NFS layer, i.e., the UIOS_FS, which is a stackable file system[3],[4]. The UIOS_FS integrates storage spaces of the multi-NAS and forms a uniform virtual shared storage space for users.

The basic idea about the UIOS_FS is as follows: (1) the system maintains a shared virtual directory tree, on which each NAS file volume is under the directory tree root, and under volumes there are shared directory nodes setup by the administrator, so each shared directory is corresponded to a NAS node respectively. And the shared

mode is very flexible in which the administrator can make any directory in the NAS file volumes be shared. The shared virtual directory is shown in Fig.2. (2) To balance loads between each file volumes in the file system, a file volume directory can be dynamically transferred to another file volume, even a subdirectory or files of a directory can be moved into another volume. These moved directories needs to be recorded onto the sharing directory tree so that users can consult them properly. Besides, the whole process can be shown to users. (3) To locate a user's request quickly, the shared virtual directory tree can build the shared directory tree in users' view, and it can separate users from concrete physical volumes. The users' shared directory tree is administrated on the basis of their identities, of which the roots are the users' nodes. The second layer of each shared directory tree is the shared directory nodes set up by the administrator. The shared directory nodes may be different shared directories in different volumes, and there are the moved directory nodes under the shared directory nodes, and the moved directory may be multilayer, as shown in Fig.3

When the UIOS_FS is concretely implemented, it maps a logic path for the users' view to the NAS physical path.

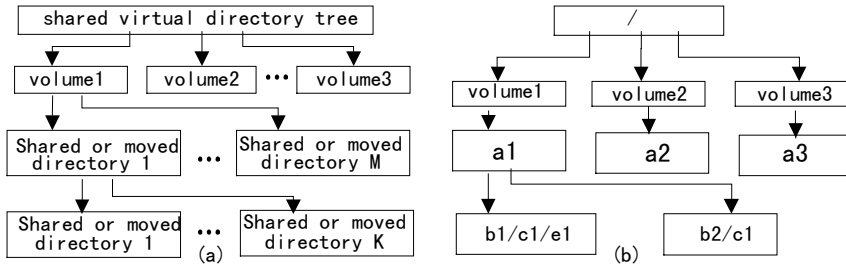


Fig. 2. The figure (a) is shared virtual directory tree, the figure (b) is shared virtual directory tree example

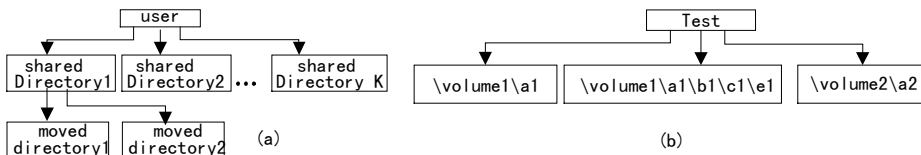


Fig. 3. The figure (a) is user directory tree, the figure (b) is user directory tree example

The shared directory tree example is shown in Fig.2(b), the shared a1, a2 and a3 directories are under volume1, volume2 and volume3. The shared b1/c1/e1 and b2/c1 directories are under a1, although the two directories are different layer at physics, they are the same layer in the shared virtual directory tree. Because the b1 and b1/c1 directories under the a1 are not shared, they are not recorded. In the Fig 2(b), if the Test user can access the shared directory a1, b1\c1\e1 and a2, and the a1 and b1\c1\e1 are under volume1, the shared directory a2 is under volume2, the Test user directory tree in Fig.3(b) is built by Fig.2(b). In the iNASC, there are two kind of

volumes, one is management volume which is located in the metadata server and each NAS node, the another is data volume which is located in each NAS node.

3.2 The Design of a Load Balancing File System

The iNASC must provide data backup and respond reading request to a number of users simultaneously. Due to each data volume capability in the iNASC is limited and the maximum I/O number a NAS can accept is finite, we design a load balancing file system, i.e., the LOADB_FS, which implements three key features: (1) Reasonable I/O scheduling. The LOADB_FS real-time records occupational status of each data volume storage space and the running process for each NAS, according to these information LOADB_FS tries to transfer backup operations to the data volume that still remains a large storage space and runs lesser processes. At one time, if a NAS has too much reading requests which overflows its limit, the LOADB_FS will automatic copy the share files read by multi - users into the another data volume which locates in another NAS and has lesser I/O loads, then these data volumes which have the same files respond to users' reading requests simultaneous. (2) The iNASC can be used in a heterogeneous environment (Unix, Windows and iSCSI client) because the capacity of data volume may change with shipping year and cost. The capacity unbalance between iNASC members may cause a concentration of file-access requests from clients. In response to these circumstances, iNASC has an autonomous rebalancing facility that moves files between data volumes automatically and dynamically without any client administration. (3) Automatic migration. When the iNASC metadata server receives an automatic migration request from clients, iNASC stops file-sharing services on the existing NAS system and mounts the file system of the existing NAS on iNASC. Then LOADB_FS traces the files-and-directories tree on the existing NAS and copies this tree to the file system on the management volume. After LOADB_FS makes the files and directories tree on the management volume, it restarts the file-sharing service for clients and moves some virtual partitions corresponding to the existing NAS to another data volume in the background.

3.3 The iSCSI Protocol

The iSCSI defines a mapping from the SCSI to the TCP/IP, namely it packs the host SCSI commands to the IP packets and transports them over the IP network. When the packets arrive at the destination node, they would be resumed to the SCSI commands, consequently, a direct and transparent transport process for the SCSI command over the IP network is realized, which integrates the SCSI and the TCP/IP protocol and realizes a non-slot connection with the storage system and the network [1],[2].

Nowadays, there are three ways for the iSCSI realization [1]. The iNASC implements the iSCSI function by way of the pure software. The iSCSI software modular includes the Initiator modular which is located at the client and the Target modular which is located at the server. Both of them are loaded to the OS as kernel state drivers. The Initiator is responsible to intercept and capture the I/O requirements handed

down by the file system and transforms them into the iSCSI data units, then sends them via the network interface card. The target is responsible for the SCSI commands and delivers them to the SCSI device adjusting to the information of the iSCSI protocol data units. After loading the Initiator modular in a client, there will appear a device named `/dev/sd*`, which can be directly mounted to the system. The Target modular is load to metadata server and each NAS on the iNASC. The Target modular has three mode to response an iSCSI request: the file I/O, the memory I/O and disk I/O mode. The iNASC mainly uses the disk I/O and memory I/O mode.

3.4 Design for Multi-user and Multi-task Parallel FTP Module

To enhance system performance, we design a multi-user, multi- task parallel FTP module. It can serve for multi-user at the same time, and each user can submit multi-task simultaneously, and all tasks are processed in parallel, thus the system resource utilization rate and the whole system performance can be improved. We deal with these parallel tasks in multi-thread mode, in which each task is corresponded to a thread; furthermore, a main thread takes charge of processing user tasks submission while timeout thread takes charge of monitoring the task processing thread to find out whether or not it is timeout. The whole FTP server is made up of three threads and two queues. The three threads are a main thread, a timeout process thread and a task processing thread. The two queues are a user queue and a user task queue. These three threads are related with each other by these two queues and all of them harmoniously work together to function as a parallel FTP server.

3.5 iNASC Manager

To maintain the manageability of a single NAS while providing a high level of scalability, the iNASC manager implements on-line reconfiguration, collaborates with UIOS_FS and the LOADB_FS, and provides metadata management. On-line reconfiguration, which can add or remove NAS nodes easily and transparently without stopping the client file-sharing services, is a strong feature of iNASC. When one of the NAS nodes in iNASC is unstable, the administrator removes this NAS node by using a Web browser. After that, iNASC automatically moves all accessible files from the unstable NAS node to the other nodes.

4 The Software Architecture for iNASC Communication

The software architecture of the iNASC communicating with client is shown in Fig. 4. When the iNASC offers the block I/O services, it uses the iSCSI technology, as shown in Figure 4 (Client 1). Concrete data read/write process is: (1) The block I/O commands (SCSI commands) sent by the application in Client 1 are encapsulated to the IP data packets via the iSCSI device driver, then transferred over the IP network; (2) When the encapsulated packets arrive at the iNASC metadata server, they are

restored to the original SCSI commands via an unpacked process, then come to the VFS layer. After being processed by the UIOS_FS and the LOADB_FS, the former I/O commands are packed again and sent to an appointed NAS via an inner network; (3) The requested data blocks are packed to the iSCSI protocol data units by the NAS, and returned to the requiring user by a high - speed IP channel.

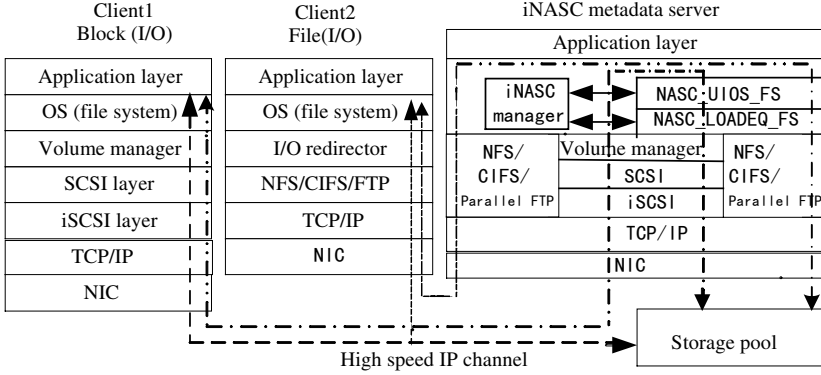


Fig. 4. The software structure for the client and metadata server communication

When iNASC offers a file I/O service, the data read/write process is almost the same as that in a traditional NAS mode. The File I/O users can communicate with the NAS by parallel FTP, and can fully utilize the Zero-copy function provided by the NAS, so the iNASC has ultra-high-throughput for the file I/O requests [1].

5 Experiment Evaluation

The experimentation uses a metadata server (host1), a PC (host2) that can send both the block I/O request and the file I/O request, and two NAS. Host1: CPU(Intel Pentium4 1.7G), memory(256MB), OS(Linux 7.1), Hard disk(IBM 60G), NIC/HBA (AGE-1000SX); Host2: CPU(Intel Pentium4 1.6G), memory(256MB), OS(Linux 7.1), Hard disk(Maxtor 40G), NIC/HBA(AGE-1000SX); NAS: CPU(Intel Pentium4 1.5G), memory(1GB), OS(Linux 7.1), Hard disk(IBM 60G), NIC/HBA(AGE-1000SX), RAID(FICS-RAID). Our experiment goal mainly is testing average response time, I/O throughput for the iNASC and its influence on the metadata server performance. The test tools are the Iometer and Bonnie++. The block I/O and file I/O is tested by the Iometer, and the metadata server performance is tested by the Bonnie++ when it load or unload the UIOS_FS and the LOADB_FS.

5.1 Experiment Result

Figures 5 show the effect of the file/block size on the throughput and the mean response time. From these figures, we can see that when the file/block size is increased, the throughput and mean response time is increased while the IO/s is decreased. As

shown in Figure 5, there is no performance difference between the block I/O requests and that of the file I/O when the block I/O requests are processed by the iSCSI and the file requests are processed by the optimized file system and a parallel FTP in NAS and metadata server.

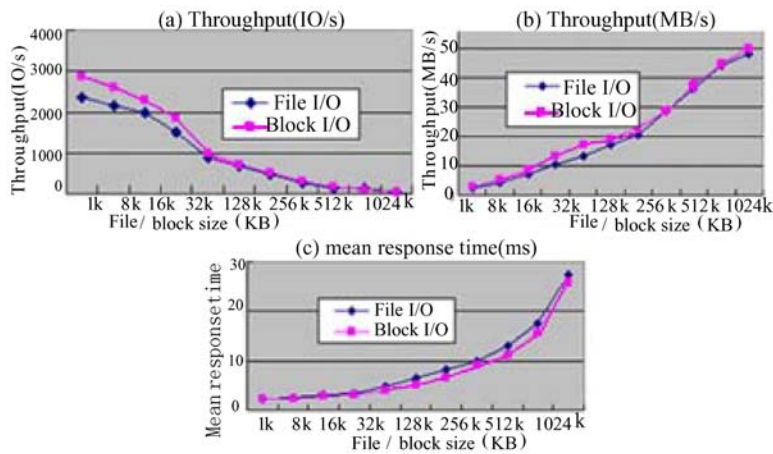


Fig. 5. The curve of average response time and I/O throughput.

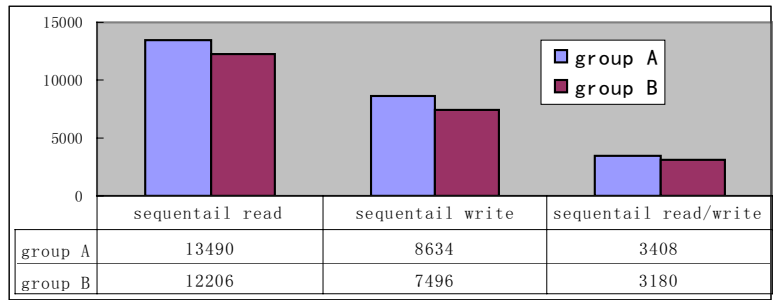


Fig. 6. File system transfer rate

The Bonnie++ can test the file system transferring speed and the CPU occupation rate in three modes, i.e., sequential reading, sequential writing, and random location. In the experiment, the file size and memory size were 200M and 50M respectively. Figure6 presents the test result of two data groups: data about the file system NFS (appointed as group A), and data about the NFSUIOS_FS+ LOADB_FS (appointed as group B).

5.2 Data Analysis

As shown in Figure 5, we have loaded the iSCSI module and the parallel FTP to the metadata server, and we have appended our products with a high-speed cache, an

intelligent pre-fetch, a distributing file system, etc. The iNASC file I/O and block I/O speed are as high as 50MB/s.

In Figure 6, group A denotes the case that has not loaded the UIOS_FS and the LOADB_FS, group B denotes the case that has loaded the two file systems. When the sequential block reads, the speed for group B has decreased 9.6% than that for group A; When the sequential block writes, the speed for group B has decreased 12.9% than that for group A; When the sequential block reads/writes, the speed for group B has decreased 11.4% than that for group A. The facts explain that the system performance will be decreased about 9.6%-12.9% when the UIOS_FS and LOADB_FS are loaded to the metadata server kernel layer, but the impact can be considered quite little when we compare the effect brought by the UIOS_FS that provides a uniform storage view to users with the effect brought by the LOADB_FS that uniformly distributes all load to each data volume.

6 Conclusion and Prospect

By storage virtualization and load balancing technology, the multi-NAS are integrated into the iNASC. Via the iSCSI, the iNASC can respond to both the block I/O request and the file I/O request; A FTP module improves the system resource utilization rate and makes the whole system perform better. In the iNASC, data copying between the NAS uses the NFS protocol, of which the configuration is very simple. Besides, we develop the UIOS_FS and the LOADB_FS with a stackable file system technology, which has little effect on the iNASC metadata server when the modules are loaded.

In the future works, we will integrate the common NAS, the iSCSI-based NAS, the block storage devices, the object storage device into a storage pool, optimize and improve each management module in the iNASC metadata server, making it to achieve a higher performance and better compatibility.

References

1. HAN Dezhi, XIE Changsheng, FU Xianglin. The design and implementation of An iSCSI-based Network Attached Storage server. *Research and Development of Computer*, 2004(1).
2. Zhou D, Xubin H. Web proxy analysis in distributed iSCSI RAIDs. *System Theory*, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on, March 14-16, 2004, Pages:225-229.
3. Yasud Y, Kawarnoto S, and so on. Concept and evaluation of X-NAS: a highly scalable NAS system. *Mass Storage Systems and Technologies*, 2003. (MSST 2003). Proceedings. 20th IEEE/11th NASA Goddard Conference on, 7-10 April 2003, Pages:219 - 227.
4. E. Zadok et.al. FiST: A Language for Stackable File Systems. *Proceedings of the Annual USENIX Technical Conference*, June, 2000.
5. Auspex whitepaper: NAS-SAN Convergence Today: New Trends in Enterprise Storage [EB/LO]. www.internetnews.com/storage/article.php/, 2004.

Design and Implementation of a Non-volatile RAM Disk in the SAN Environment

Ji-wu Shu, Bing Yu, and Rui Yan

Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China
shujw@tsinghua.edu.cn

Abstract. The mechanical nature of the magnetic disks limits the possibility of significant improvement of the I/O performance of the magnetic disk storage systems currently in use. The use of magnetic disk storage system has become an obstacle to the performance development of critical applications. This paper describes an implementation of a remote non-volatile RAM disk (abbreviated as NVDisk) over Fiber Channel network. Read and write latencies are drastically reduced and thus the I/O performance of the storage system is improved by order of magnitudes. We implemented an NVDisk target driver to provide full standard SCSI command set support, so a virtual disk can be constructed for use in the storage area network. NVDisk does not engage the foreground server's CPU and main memory resources, so it can undertake extremely heavy workloads. In addition, we implemented a Virtual Disk (VD) module in the Linux kernel, which used a memory pool and backup disks to form a virtual transparent appliance and achieved the encapsulation of the ramdisk. With this, snapshot-based online backup mechanisms can be carried out. The whole system was built in the FC SAN environment, so the NVDisk is fine scalable and can be shared easily between servers.

1 Background

Traditional random access storage systems use magnetic disks as recording media to prevent data loss at system failure. Magnetic disks have long dominated the storage market because of their relatively low media cost, high capacity and reasonable performance. However, due to their mechanical nature, the possible I/O performance improvement of the current magnetic disk storage systems is very limited. With the intense growing pressure of I/O bandwidth needs, the use of magnetic disk storage system has become an obstacle to the performance development of critical applications. Techniques have been developed to alleviate the I/O problem, such as the read cache, non-volatile write cache, RAID (to improve available I/O rates by reading in parallel from an array of disks [1]), and disk cache disk (DCD[2]) to improve random write performance by adding a journal disk. The DCD first writes on the journal disk sequentially and then flushes the journal to the normal disk asynchronously.

On the other hand, the unit price of DRAM is going down because of its increasing density. The unit price of a magnetic disk is approximately 1\$/GB

and DRAM is approximately 0.2\$/MB, a price difference of about 200 times. However, the average latency of magnetic disks is about 10ms, which is far longer than DRAM's 10ns latency. The difference between them is about 106. Therefore, DRAM is quite cost effective for high performance applications. In addition, the seeking and rotational latency of magnetic disks is the main cause for its poor performance, whereas DRAM doesn't have this drawback. These two latencies can also cause performance fluctuation and a sharp drop in performance when the workload increases. This is characterized by an unsteady number of IO operations performed per second and the average is quite low, which is far from the requirements of critical applications. Using the proposed NVDisk as a cache disk for special functionalities can improve overall performance significantly. For example, using the NVDisk as a non-volatile write cache can remarkably reduce synchronous write latency, which is important in journal filesystems [3] and transaction commitments.

This paper describes an implementation of a remote non-volatile disk (NVDisk) for FC SAN. Read and write latencies are drastically reduced and thus the I/O performance of the storage system is improved by order of magnitudes. We implemented an NVDisk target driver to provide full standard SCSI command set support, so the virtual disk could be constructed for use in a storage area network. Compared with traditional ramdisk implemented in operating systems, NVDisk does not engage the foreground server's CPU and main memory resources, so it can undertake a very heavy workload. In addition, we implemented a Virtual Disk (VD) module in the Linux kernel, which made the memory pool and backup disks into a virtual transparent appliance, and accomplished the encapsulation of the ramdisk. With this, it is possible to provide snapshot-based online backup mechanisms. The NVDisk I/O node is protected with UPS, and has data recovery functionalities at start up. The whole system was built in the FC SAN environment, so the NVDisk is fine scalable and can be shared easily between the servers.

2 Related Work

There are several research orientations relevant to high performance storage systems, the most relevant ones are:

NVRAM (Non-volatile RAM) Technology: any storage appliance that consists of battery-backed low power SRAM or a small amount of SRAM on non-volatile FLASH chips. The main disadvantage of these kinds of products is their high price (four to ten times as much as volatile DRAM [4]) and relatively small capacity. For now, NVRAM is mostly used as a Firmware carrier to replace the old fashioned EEPROM and also as a non-volatile write cache built into magnetic disks.

Flash Memory Technology: a kind of reprogrammable EEPROM, addressed by block instead of byte. It has a price comparable to DRAM of the same size, but has very high write latency and limited write/erase cycles, and can only be used in WORM (write-once-read-many) applications. eNvy [5] uses Flash chips and a

small amount of battery-backed low power SRAM to build a high performance massive non-volatile storage system. A series of algorithms and mechanisms were developed to avoid the main drawbacks of the Flash memory, such as high write latency and limited read/write cycles.

Remote Ramdisk: a method to make a remote LAN server's main memory serve as a local ramdisk. NRD [6] is a technology that uses the main memory of the remote server as a local block device. This project is implemented among the NOW cluster nodes. Mirroring and parity algorithms are used to ensure redundancy. Pnevmatikatos et al. [4] presented a software-based "NVRAM" storage system that uses a remote LAN server's volatile main memory as non-volatile storage. Multi-node redundant mechanisms are used to avoid data loss due to power failure of a single node.

MRAM (Magnetic RAM): a recent hot spot in the field of non-volatile storage[7]. MRAM is as fast as SRAM and does not need electronic power to keep information durable. However, this attempt is still at the stage of prototype in the labs. Desikan et al. [8] evaluated MRAM based storage systems and pointed out the advantages, which include low access latency comparable to that of SRAM, low read power consumption and high density. Disadvantages such as relatively high power consumption and relatively low speed while writing are also indicated. In [9], non-volatile MRAM is used to store the real-time compressed journal information of a certain file system, and thus the performance is improved.

Companies such as Texas Memory systems and Curtis have developed products called **Solid-state disks**, which consist of battery-backed SDRAM modules. Control logic is implemented with ASIC chips, so the performance is quite high, though unfortunately it costs much.

Our software-based approach is easier to implement, more flexible for migrating and upgrading, and also costs little. The problems of lower performance and a higher failure rate can be avoided by load balance and redundant mechanisms.

3 The NVDisk Architecture

3.1 Hardware Architecture

NVDisk is based on the TH-MSNS (Mass Storage Network System)[10][11] SAN system developed by ourselves. The main components of TH-MSNS are: a console node, I/O processing nodes, high density magnetic disk arrays, Fiber Channel switches and Fiber Channel interconnectors and Ethernet interconnectors. The processors on the I/O nodes are classified into two kinds: FCP processors and storage processors, functioning respectively. In short, the whole system consists of target nodes (namely I/O nodes), initiator nodes (namely foreground server nodes) and the management node (namely the console node), as shown in figure 1. A pair of I/O nodes acts as the SCSI target, and both the I/O nodes and the foreground server nodes are interconnected with a 2Gbps Fiber Channel via FC HBA adapters. I/O nodes are supplied with UPS to avoid random power failure. NVDisk is implemented on the I/O nodes.

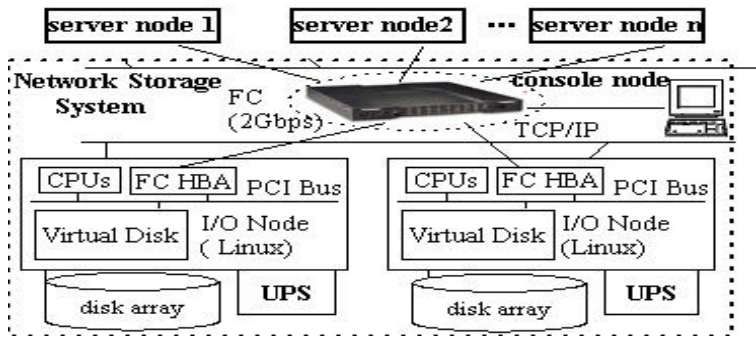


Fig. 1. NVDisk architecture based on TH-MSNS

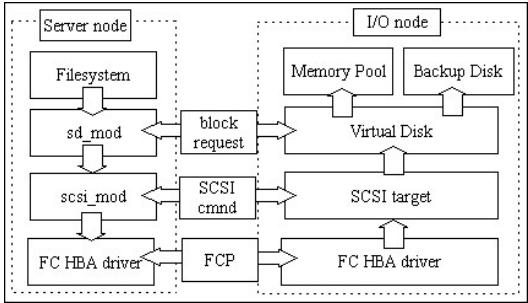


Fig. 2. The data write path

3.2 Software Architecture

The data write path is shown in figure 2.

The file system of the initiators submits I/O requests according to the users' demand. These I/O requests are re-scheduled and consolidated in the block layer, and then translated into SCSI commands and handed over to the SCSI layer modules. The SCSI layer modules pass them to the FC HBA driver. Then the SCSI commands are encapsulated in the FC frames and transmitted to the remote target node by the host FC HBA adapter. After the target receives the FC frames, the SCSI commands are recovered and handed to the SCSI target simulator. The SCSI target simulator produces the raw I/O requests and informs the Virtual Disk to complete the real I/O operations and finally returns the result layer by layer back.

From the foreground server's view, the FC HBA card driver detects the disks on the storage network, sends them to the SCSI layer and then makes them up into a usable block device for the filesystems.

4 NVDisk Implementation

In order to embed hybrid functionalities into NVDisk, we implemented a VD (Virtual Disk) module. An online backup mechanism based on snapshot and a redundant method with dual journals is achieved in the VD module. The target node is protected with UPS to prevent power failure, and the data can be recovered from backup magnetic disks during starting up.

4.1 SCSI Target Implementation

In [10][11] a mature SCSI target simulator is implemented. Our NVDisk target driver is developed upon that foundation.

The target simulator can hand the I/O requests over to the VD module by parsing the SCSI commands. The process of SCSI commands is explained in ref[10][11].

4.2 Basic Functions of the Virtual Disk

We reserved a great deal of consecutive high physical memory during the system initialization process to build up a big memory pool and only provide enough physical memory for the operating system. The upper memory probe algorithm is BEB (binary exponential backoff) with the convergence precision controlled to 1k. The available space of the memory pool is divided into pages, and the usage is managed with a bitmap. The VD object is thus formed to provide a uniform interface for the lower layers.

It is possible to either allocate buffers while dealing with DMA requests and copy the contents of the VD onto them or just hand the physical pages of the VD directly as DMA buffers. The latter can be called zero copy and zero allocation in figure 3.

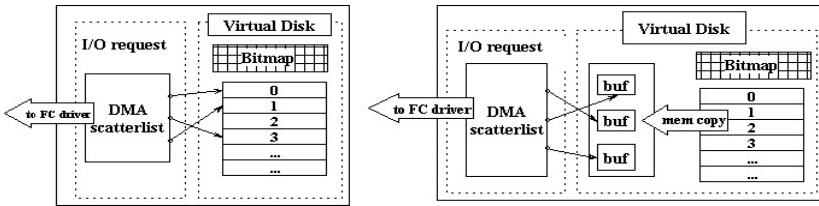


Fig. 3. Non-zero copy and Zero copy

4.3 Backup and Recovery

Software failure and human mistakes are inevitable but should be bearable in a distributed system. Our NVDisk initiator node can be rebooted after a crash and then a filesystem recovery can be performed with user space utilities or with some

rollback or commit operations according to the filesystem journal. In this way, the system can be consistent again. Correspondingly, the NVDisk target node is protected by UPS, so the main cause of node failure is system crashes (including software and hardware crashes). The primary redundant measure employed by the NVDisk to is asynchronously flush the contents of the VD to backup magnetic disks.

In figure 4(a) each VD object has two backup magnetic disks. A kernel committer thread flushes data periodically to the backup disks according to the bit changes in the bitmap. While the system is starting up, the VD object fetches all the data from the backup disks and reforms itself. In order to keep the contents of the backup disks consistent with the VD object at some specific point in time, a backup mechanism like the snapshot is used.

After the I/O requests are successfully written into the pages, the corresponding bit in the bitmap is marked “dirty”. When the committer thread starts, first the zero copy mechanism is disabled and a snapshot I/O buffer queue is set up to hold all the I/O requests after this point in time. Then all the pages marked dirty are flushed into the corresponding blocks of the backup disks. The combination of these operations is called a disk transaction. After the disk transaction is complete, all contents in the snapshot I/O buffer are flushed into the memory pool, and the corresponding bits in the bitmap are set. Finally, the zero copy mechanism is enabled again.

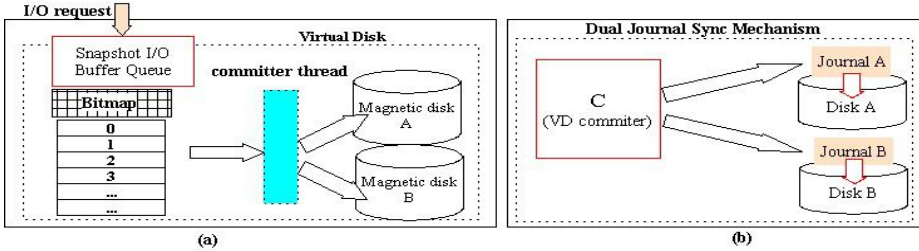


Fig. 4. The working committer thread/Dual journal, dual backup disk sync mechanism

In order to resolve the problem of inconsistency after system failure, we proposed a dual-journal-dual-backup-disk redundancy mechanism in figure 4(b).

For convenience, we assigned “A” to designate disk A, and “B” for disk B, “A(L)” for the journal on disk A and “B(L)” for the journal on disk B. The VD committer thread is represented by “C”. The process for each disk transaction is as follows:

At the beginning, A is in the consistent state of point T(1) in time, A(L) logs the increment $\Delta(0)$ from point T(0) to point T(1); B is in the consistent state of point T(1) in time, and B(L) logs the increment $\Delta(1)$ from point T(1) to T(2).

At the time of T(3), C flushes the increment $\Delta(2)$ from T(2) to T(3) to A. The flush policy is as shown below:

Step1: $B(L) \xrightarrow{\Delta(1)} A$:at this time B is in the consistent state of time point T(2), while A is not consistent.

Step2: $C \xrightarrow{\Delta(2)} A(L)$:at this time A and B are in the consistent state of time point T(2), while A(L) is not consistent.

Step3: $A(L) \xrightarrow{\Delta(2)} A$:at this time B is in the consistent state of time point T(2), while A is not consistent.

After the flush procedure, we can ensure that B is in the consistent state of time point T(2), B(L) logs the increment $\Delta(1)$ from T(1) to T(2), A is in the consistent state of time point T(3), and A(L) logs the increment $\Delta(2)$ from T(2) to T(3). After this, the disk transaction is complete. The process is repeated with the roles switched, and then it is ready for the next disk transaction.

Each step submits a status mark in the journal when finished. If there is no mark logged, it shows that the last write operation failed at a certain step. The inconsistent disk will sync with the consistent one when the NVDisk starts up. If it failed during step 1, B(L) is flushed to A, and then A and B are in the consistent state of time point T(2). If it failed during step 2, A(L) is invalidated, and then A and B are in the consistent state of time point T(2). If it failed during step 2, A(L) is flushed to A, and then A is in the consistent state of time point T(3), and B in the consistent state of time point T(2).

5 Testing and Performance Evaluation

Tests were designed and performed to evaluate and analyze the performance of the NVDisk system. Table 1 shows configurations of the hosts and storage nodes for our test.

Table 1. Test configuration of hosts and storage node

A: Local and remote Storage node and their storage subsystems/Host		
CPU	Intel Xeon 2.4GHz $\times$ 1	Intel Xeon 2.4GHz $\times$ 1
Memory	1G	1G
OS	Linux with kernel 2.4.25-lck1	Linux with kernel 2.4.25-lck1
FC HBA	Emulex LP982 (Initiator,2Gb/s)	Emulex LP982(Initiator)
	Qlogic ISP 2300 (Target,2Gb/s)	
RAID Controller	Adaptec U160 RAID 2110S	
SCSI Disks	Seagate (73GB 10K) $\times$ 7,JBOD	

The standard disk I/O testing kit Iometer by Intel Corporation [12] was used. In order to bypass the operating system's block layer buffer and block scheduler, we modified the source and used DIRECT_IO, which is quite different from the normal version. Therefore, our testing results show the actual performance of the disks.

5.1 Comparison of Performance Under Heavy Workload

Three foreground server nodes were used in the tests, each with two Iometer instances. Each instance had two worker threads, so there was a total of 12 concurrent worker threads. They could submit 100% random I/O requests onto the same disk. In this circumstance, the workload represented by IOPS (I/O operations per second) reached its utmost limit, so the average latency and throughput under the highest workload could be measured, as shown below in figure 5(a).

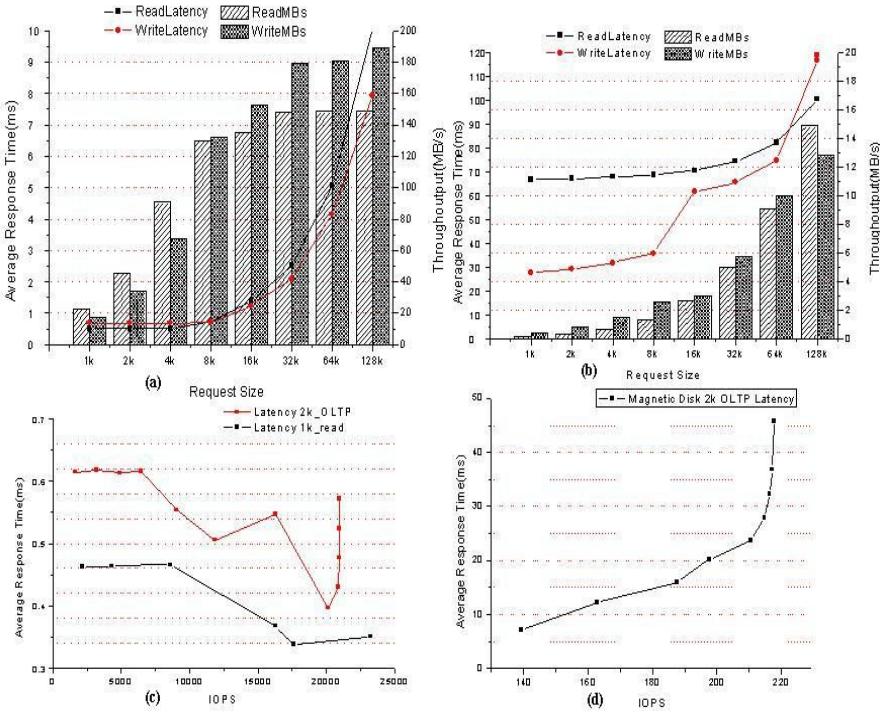


Fig. 5. NVDisk throughput and latency under heavy workloads

From the figure 5(a), we can see that the average response time rises sharply when the request block size is bigger than 16k bytes. At the same time, throughput rises slowly while IOPS drops. This indicates that the 2G bandwidth of the Fiber Channel becomes the bottleneck of data flow in our system. Therefore data with request size smaller than 16k can reflect the real I/O processing capability of the NVDisk target:

To make a comparison, we added a 10000 rpm SCSI magnetic disk to the same I/O target node and exported to the storage network. With the same testing methods as above, we obtained the results shown below in figure 5(b).

The corresponding IOPS data is showed in Table 2:

Table 2. Magnetic disk and NVDisk utmost IOPS under various request sizes

Request size	1k	2k	4k	8k	1k	2k	4k	8k
Read IOPS	179	178	176	174	23414	23417	23212	16638
Write IOPS	430	412	375	334	17285	17320	17319	16875

Based on these results, it is clear that NVDisk has excellent performance under heavy workloads. Its read IOPS is about 100 times more than a magnetic disk, while its write IOPS is about 40 times more. It has much better throughput and latency results than a magnetic disk.

5.2 NVDisk Performance Curve Under Various Workloads

We used the access pattern defined by Iometer of Intel Corp. to simulate the OLTP workload, which had 2k totally random request blocks and 67% read requests with 33% write requests. Another series of tests with 1k totally random read request size was also performed. The results were taken under various workloads represented by IOPS in figure 5(c).

Similarly, a 10k rpm SCSI magnetic disk on the storage network was used for comparison. A test of the 2k OLTP access pattern was performed in figure 5(d).

The results show that the average response time of the NVDisk is stable when the workload is low, and fluctuates when the workload gets high. This is related to the processing capability of the I/O target node and the FC HBA adapters. The numerical value during the stable phase shows the average read latency is 0.46ms or so and the average latency of the 2k OLTP access pattern is about 0.61ms. Under the OLTP workload, the NVDisk can reach a high IOPS and at the same time preserve low latencies compared to a magnetic disk.

6 Conclusions and Future Work

High performance storage appliances are playing important roles in applications with extraordinary requirements. The NVDisk was designed according to such demands. It provides high performance with an SDRAM based storage pool, high reliability with snapshot based dual-journals-dual-disks backup mechanisms, and high scalability with the natural sharability in the storage network.

As for future work, we are currently investigating the feasibility of porting the NVDisk to real-time embedded systems. The objective is to further increase the overall capacity and performance. Hybrid multi-target redundant methods are also to be developed.

Acknowledgements

The work described in this paper was supported by the National High-Tech Research and Development Plan of China under Grant No. 2001AA111110.

References

1. Peter M. Chen, Edward K. Lee, Garth A. Gibson, et al., Patterson. RAID: High-Performance, Reliable Secondary Storage. *ACM Computing Surveys*. vol.26, no 2, pp. 145-185, 1994.
2. Hu Y. M., Yang Q.. DCD Disk Caching Disk - A New Approach for Boosting IO Performance. The 23rd International Symposium on Computer Architecture, May 22-24 1996, Philadelphia, Pennsylvania.
3. Baker, M., Asami, S., et al., Non-Volatile Memory for Fast, Reliable File Systems. Proceedings of Proceedings of the 5th Symposium on Architecture Support for Programming Languages and .Operating Systems, pages 10-22, October 1992.
4. Dionisios P., Evangelos P. M., Gregory Maglis, Sotiris Ioannidis. On Using Network RAM as a non-volatile Buffer (1998). *Cluster Computing, Special Issue on I/O in Shared-Storage Clusters*, 2(4), pp. 295-303, 1999, Baltzer Science Publishers.
5. Michael W., Willy Z. eNvy - A Non-Volatile, Main Memory Storage System . Proceedings of the Sixth Symposium on Architectural Support for Programming Languages and Operating Systems, pp. 86-97, October 1994.
6. Michail D. F. and Evangelos P. M.. The Network RamDisk - Using Remote Memory on Heterogeneous NOWs. *Cluster Computing, Special Issue on I/O in Shared-Storage Clusters*, 2(4), pp. 281-293, 1999, Baltzer Science Publishers.
7. Ethan L. M., Scott A. B., Darrell L., HeRMES: High-performance reliable MRAM-enabled storage. In *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, pages 83-87, Schloss Elmau, Germany, May 2001.
8. R. Desikan, S. W. Keckler, and D. Burger. Assessment of MRAM Technology Characteristics and Architectures Technical Report TR-01-36, Department of Computer Sciences, University of Texas at Austin, 2001.
9. Nathan K., Ethan L. M, and Karl S. B. Measuring the compressibility of metadata and small files for disk/NVRAM hybrid storage systems. Technical Report UCSC-CRL-03-04, Storage Systems Research Center, University of California, Santa Cruz, July 2003.
10. Shu J.W., Yao J., et al., Highly efficient FC-SAN based on load stream, Xingming Zhou, Stefan Jahnichen, Ming Xu, and Jiannong Cao eds., *The Fifth International Workshop on Advanced Parallel Processing Technologies, LNCS 2834*, pp.31-40, 2003, Springer-Verlag Berlin Heidelberg printed
11. Shu J.W., Technical Report: Design and Implementation of TH-MSNS (In Chinese), Computer Science Department, Tsinghua University, P.R.China, 2003, <http://storage.cs.tsinghua.edu.cn/>
12. Jerry S. Iometer: The I/O Performance Analysis Tool for Servers. <http://www.intel.com/design/servers/devtools/iometer/index.htm>

Engineering Web Storage Servers Using Session Management*

Min Qu, Yafei Dai, and Yang Zhao

CNDS Lab, Peking University, Beijing, China
{qm, dyf, zhaoyang}@net.pku.edu.cn
<http://net.pku.edu.cn>

Abstract. With the Web-based distributed file storage systems increasingly being used for file storage and sharing, there is a growing need to provide high level of availability and quality of services. In this paper, we discuss the feasibility of introducing session management into layered distributed file storage systems. We improve the traditional *client - Web server - storage server* infrastructure by inserting session management layer into web servers. We also propose the strategy of leveraging session management to implement session migration and service continuation in a completely client-transparent fashion. Performance evaluation on the prototype implementation demonstrates that our approach is efficient and the overhead is reasonably small.

1 Introduction

A vast majority of today's Internet services are built over HTTP, the standard Internet application layer protocol. The rapid development of HTTP-based web applications has led to increased demands by its users with respect to both availability and quality of services delivered over internetworks. Providing desired level of availability and quality of services in Web-based distributed file storage systems has become a big challenge.

In distributed file storage systems, the main user operations are file storage and retrieval, which lasts for a relatively long time. Network congestion or server failure may impose large negative impact on system availability.

The traditional *client - Web server - storage server* infrastructure of distributed file storage systems can not provide connection failure tolerance. If the back-end machine crashes or the connection fails, all the connections to the web server break and all clients get disconnected from the server. Even if another backup server exists, a new connection has to be established between each client and the web server, and lost packets have to be determined and retransmitted. If the web server is able to dynamically migrate connection to another storage server to provide uninterrupted and undegraded service despite the connection failure, or efficiently checkpoint the connection's state and recover when service is available, system availability may be tremendously improved.

In an attempt to solve these problems, we propose a solution based on session management. A session [1] is a durable, long-term relationship between application end

* This work is supported by National Natural Science Foundation of China No. 60303002.

end points that may span multiple network connections and application transactions. Session may be deployed to provide enhanced services useful in some applications such as dialogue control and synchronization. Leveraging session management to achieve high availability and quality of service may be a reasonable and feasible choice.

In this paper, we discuss the feasibility of introducing session management into layered distributed file storage systems. We improve the traditional client - Web server - storage server infrastructure by inserting session management layer into web servers. We also propose the strategy of leveraging session management to implement session migration and service continuation.

The remainder of this paper is structured as follows. Section 2 presents the architecture of session oriented distributed file storage system. Section 3 introduces the session management idea. Section 4 gives evaluations, analyzes overhead and latency imposed by session management layer and the improved recovery time. Section 5 introduces related work in this area. Then section 6 concludes the paper.

2 System Model

The traditional web-based distributed file storage systems consist of three components: client, HTTP server, and storage servers. HTTP server consists of two parts: HTTP Daemon and Storage Connector. The first part corresponds to the front-end processes that accept the client connection for service and return responds. The second part corresponds to other back-end processes that participate in file storage and retrieval. This structure, although straightforward in implementation, has the drawback that it can not overcome the limitation of TCP protocol. TCP is the most popular transport layer protocol for constructing distributed applications over the Internet [3]. It has been used to construct several commonly used applications and protocols, including HTTP. However, an import feature that TCP does not provide is server fault tolerance. The connection-oriented nature [4] of TCP, along with its endpoint naming scheme based on IP addresses, creates an implicit binding between a service and the IP address of a server providing it, throughout the lifetime of a connection. This makes the client prone to all adverse conditions that may affect the server endpoint or the internetwork, after the connection is established.

In order to solve this problem and provide highly available services, we introduce session management into this system model, as in figure 1. Session management layer is inserted into HTTP server between HTTP Daemon and Storage Connector. This layer is responsible of connection management and state maintenance, which is functionally similar to the Session Layer in the ISO OSI reference model [2]. The session management layer sends and receives data from the neighbor layers, monitors the connection states and detects failures. In case of catching exceptions, it does some operations based on current session state to remedy the service, not simply returns exceptions to the client. This layer does not influence the implementation of the client and storage servers.

The detail system architecture is illustrated in figure 2. HTTP server consists of four parts: UI, HTTP Session Manager, FTP Session Manager and FTP Channel Handler. They each have independent functions. The client establishes HTTP connections

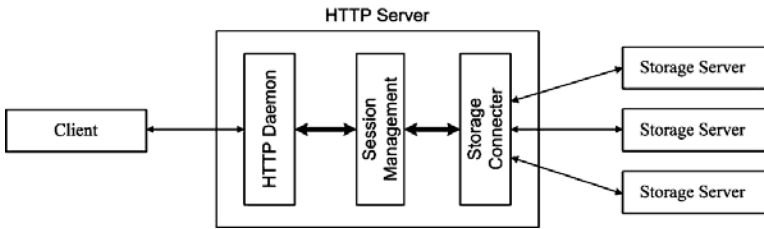


Fig. 1. The Session Management Layer inserted into HTTP Server.

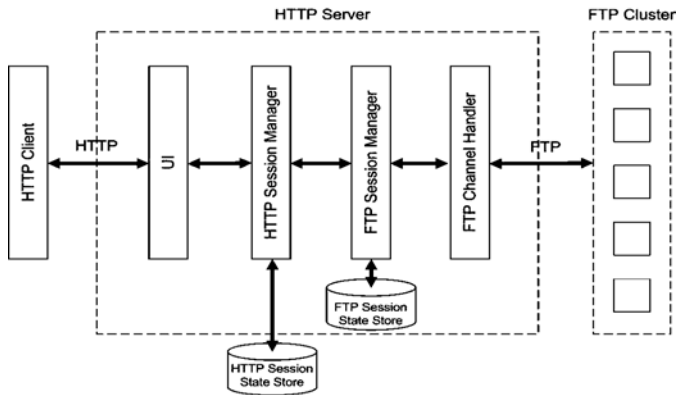


Fig. 2. System Architecture.

to HTTP Server and UI interacts with the client. HTTP Session Manager maintains session state corresponds to HTTP session, which is closely related to the front-end connections. FTP session manager is responsible for session state maintenance corresponds to FTP session, which is closely related to the back-end connections. FTP Channel Handler establishes FTP connections with FTP servers, sends requests and receives responses.

Two kinds of session state are involved in this system: HTTP session state and FTP session state, which are stored separately because of the difference in their characteristics. HTTP Session State Store saves HTTP session states and FTP Session State Store holds FTP session states. HTTP Session state and FTP Session state are located on different layers of the HTTP server, and are responsible for service continuation and session migration respectively. What’s more, HTTP session state is shared by multi HTTP servers. As its lifecycle is not restricted to a single server, it can be reinstated after disconnections and HTTP server failures. FTP session state is local to single FTP server and can not survive HTTP server failures.

This design of session management layer brings about the following benefits:

- No influence to the front-end and the back-end. It can be implemented in the manner of user transparency.
- Separation of HTTP session state and FTP session state. Deploying applicable management strategy for each state ensures high performance.

- The shared HTTP Session State Store guarantees scalability. New HTTP servers join in system easily. The Session Management Layer does not render the HTTP server a bottleneck.

3 Session Management

Session management layer supports two important capabilities: session migration and service continuation. In this section, we introduce the session management idea, describe the mechanism of session migration and service continuation.

3.1 Session Migration

Session migration is to migrate and recover service from the failed node to another working node. The implementation of session migration is based on three preconditions: a). Server pool, which is a pool of similar servers cooperate in sustaining a service by migration of connections within the pool, is efficiently established. b). Sessions are independent of each other. c). Session state is well defined and maintained.

The server pool is established by file replication. The HTTP server runs a low-priority background task to backup the newly stored files on another node based on some load-balancing algorithms. Policy files log the relationships between files and their backups. Those nodes that involved in the same relationships are located at the same server pool.

The migration process, shown in Figure 3, ensures that another server in the server pool resumes service while the back-end connection fails due to storage server failure or network congestion, without freezing or otherwise disrupting the traffic on the connection, so as to provide uninterrupted delivery of storage services.

The implementation of session migration relies on the support of FTP session manager in the HTTP server, which holds FTP connection state and monitors the connections. In case of FTP connection failure, FTP session manager catches connection exceptions and migrates the live connection to another available node.

3.2 Service Continuation

Different from session migration whose responsibility is to deal with back-end failures as has mentioned above, service continuation is proposed to treat with HTTP connection failures and HTTP server failures. The two main tasks of service continuation are checkpointing and fault-recovery.

HTTP session manager logs the intermediate state of HTTP connections in HTTP session state store by the checkpointing mechanism. The state, including session identifier, user name, file name, the flag, the uploaded or downloaded amount etc, is held in database and can be reinstate after failures. Using the above state, we can enable a request to continue downloading a file after the transfer is terminated.

An additional benefit that the server-side session state maintenance brings about is its support for personal mobility [8], [9]. For example, a user may start uploading her

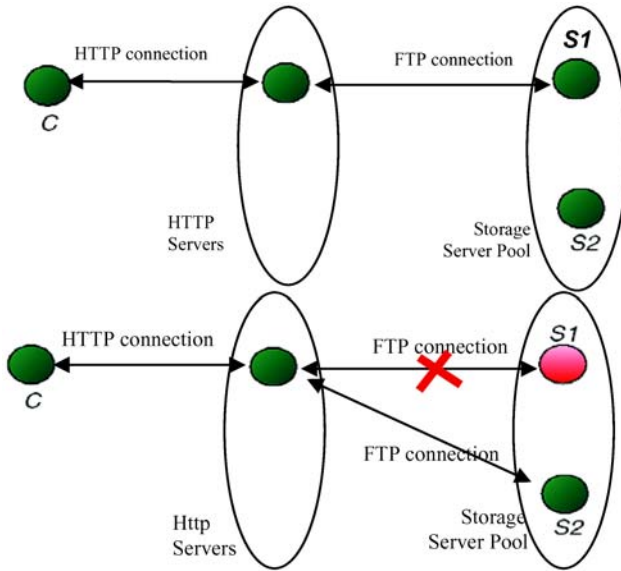


Fig. 3. Connection Migration Process. (a) Client C establishes HTTP Connection with HTTP Server M. M selects S1 to provide storage service for C. (b) While the FTP connection between M and S1 fails due to S1 failure or network congestion, M transparently migrates the connection to S2.

file on a PDA, and continues uploading on her desktop PC when she arrives at her office.

In this mechanism, session state is maintained on server side, which is different from some software that support multithread downloading with resuming capability, such as NetAnts [6] or CuteFTP [7]. In those software, connection state is logged by user side and the implementation of service continuation is relied on the support of client-side software, which is an additional requirement and adds new limitations to system.

4 Evaluation

In order to test the feasibility of our approach, we implement the prototype of a session oriented distributed file storage system. This section presents the results of performance evaluation on this prototype system.

This test is done on a shared 100-Mbps Ethernet segment. The HTTP server is an Intel 844-Mhz P3 running Apache Tomcat 5.0. The HTTP session state store is implemented by MySQL on a separate machine. One HTTP server and six FTP servers are included.

Overhead of session management layer is defined as the time consumption of accessing the shared database for state access and update, including network latency and database processing time, as shown in figure 4. Each point represents the occurrence frequency of the corresponding time consumption. The overhead is mainly distributed from 30 to 60 msec, which we think is extraordinarily small.

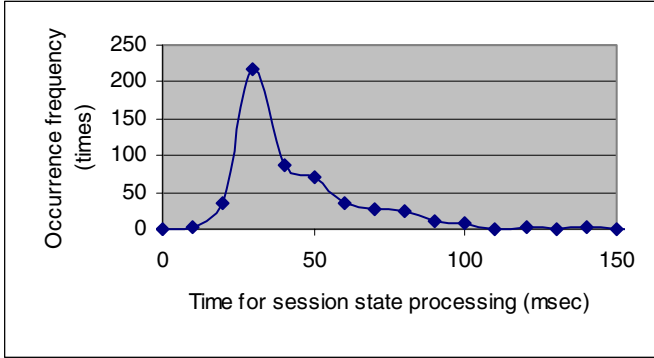


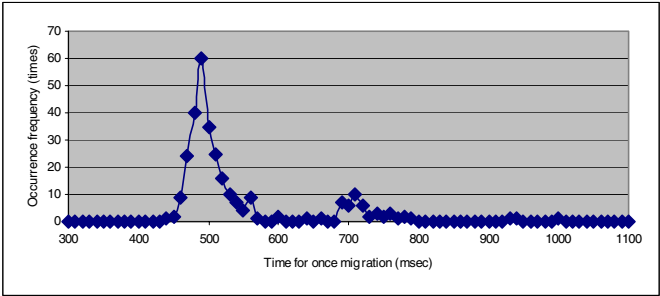
Fig. 4. Session management layer overhead.

An important parameter to evaluate system performance is the Mean Time to Repair (MTTR). A low latency failure detection and recovery mechanism that can quickly identify failure occurrences and recover on another node may results in higher availability. In this prototype system, the back-end failures recovery is achieved by session migration. This recovery time, shown in figure 5, includes the time for selecting a new node from the server pool, establishing a new connection, and preparing stream transfer from the mid-interrupted point. Each point in this figure represents the occurrence frequency of the corresponding recovery time which varies along with the migration times. Figure (a) is recover time for migrating once, (b) for twice, and (c) for three times, Each recovery time is represent 490 msec, 640 msec, and 690 msec. In most cases, the recovery time spans less than 1 second, which is an extremely small recovery time.

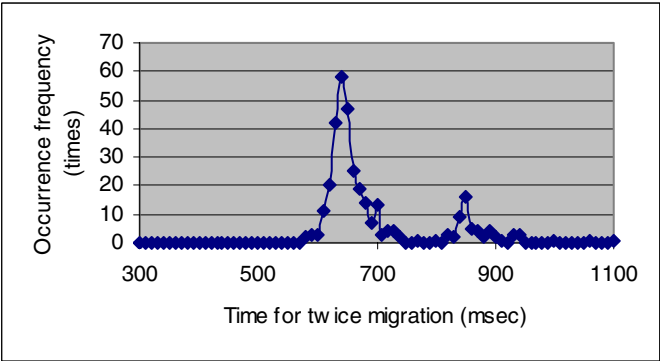
These measurements show that session oriented architecture for the development and deployment of session-layer functionality can significantly assist in achieving highly-available storage services with small overhead. Our scheme is a reasonable and feasible choice.

5 Related Work

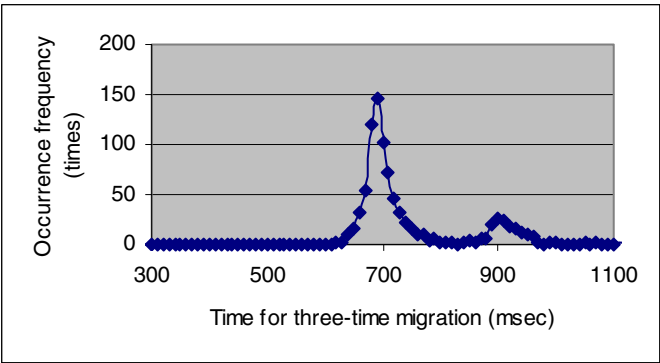
A number of approaches for achieving fault tolerance and high availability have been investigated over recent years. One approach is to insert a layer of software at the transport layer, such as FT-TCP [10]. FT-TCP uses the two wrappers put around the TCP server code to forward TCP byte stream to a logger where server state is stored. The drawback in this approach is the large failover time including failure detection, start time of backup server, and reinstatement time of server state. A second approach is to redirect all TCP connections to a proxy between the client and server, such as [11]. The primary drawback is overhead and bottleneck of the proxy. A third approach is to re-design TCP protocol to enable the capability of checkpointing, such as SC [12]. The main drawback is the new TCP implementation. Whenever the standard TCP implementation is changed, the re-designed protocol must be re-implemented.



(a) Recovery time for once migration.



(b) Recovery time for twice migration.



(c) Recovery time for three-time migration.

Fig. 5. Recovery time.

6 Conclusion

This paper discusses the feasibility of introducing session management into layered distributed file storage systems. In order to overcome the drawback of the traditional *client – Web server – storage server* infrastructure of the distributed file storage sys-

tems, we insert a session management layer into web servers to enable the capability of session migration and service continuation.

We present the session oriented system model and discusses the session management strategy, including the mechanism of session migration and service continuation. In order to test the feasibility of our approach, we implement a prototype system. Measurements show that the overhead of our scheme is reasonably small. This architecture with the development and deployment of session-layer functionality significantly assists in improve highly-available storage services.

References

1. M. A. C. Snoeren. A Session-Based Architecture for Internet Mobility. PhD thesis, Massachusetts Institute of Technology, Dec. 2002.
2. Andrew S Tanenbaum. Computer Networks. Prentice Hall, Inc., third edition, 1996.
3. M. Marwah, S. Mishra, and C. Fetzer. TCP Server Fault Tolerance Using Connection Migration to a Backup Server. In Proc. DSN '03.
4. F. Sultan, A. Bohra, and L. Iftode. Service Continuations: An Operating System Mechanism for Dynamic Migration of Internet Service Sessions.
<http://paul.rutgers.edu/~sultan/application/papers/srds03.pdf>
5. B. C. Ling, A. Fox. A Self-Tuning, Self-Protecting, Self-Healing Session State Management Layer. AMS'03.
6. NetAnts homepage, <http://www.netants.com/>.
7. Cute FTP homepage, <http://www.cuteftp.com/>.
8. P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, X. Zhao, and M. Baker. The mobile people architecture. ACM Mobile Computing and Communications Review (MC2R), 3(3):36–42, Jul. 1999.
9. H. Schulzrinne. Personal mobility for multimedia services in the Internet. In Proc. European workshop on Interactive Distributed Multimedia Systems and Services, pp. 143–161, Mar. 1996.
10. L. Alvisi, T. Bressoud, A. El-Khashab, K. Marzullo, and D. Zagorodnov. Wrapping Server-Side TCP to Mask Connection Failures. In Proc. IEEE INFOCOMM '01, Apr. 2001.
11. D. Maltz and P. Bhagwat. TCP Splicing for Application Layer Proxy Performance, Mar. 1998. IBM RC 21139.
12. F. Sultan, A. Bohra, and L. Iftode. Service Continuations: An Operating System Mechanism for Dynamic Migration of Internet Service Sessions.
<http://paul.rutgers.edu/~sultan/application/papers/srds03.pdf>

Topology and Resource Discovery in Peer-to-Peer Overlay Networks*

Dongsheng Li, Nong Xiao, and Xicheng Lu

School of Computer, National University of Defense Technology
Changsha 410073, P.R. China
leedongsh@hotmail.com

Abstract. Peer-to-Peer networks have attracted significant attention these days. The paper firstly introduce the characteristics and challenges of P2P networks and then surveys four categories of P2P topologies: centralized topology, decentralized unstructured topology, decentralized structured topology and partially decentralized topology. The characteristics, advantages and disadvantages and current researches of the four topologies are discussed. Many open problems and their recent developments are analyzed thoroughly.

1 Introduction

In recent years, peer-to-peer (P2P) overlay network has become a promising technique to take advantage of vast number of resources on the Internet. In P2P networks, each node (peer) can act as both client and server with equal capability. Peers can exchange information directly with each other and perform certain critical function coordinately in a decentralized manner.

P2P network has attracted significant attention in both industry and academic research. There are many applications of peer-to-peer overlay networks, such as distributed computing (e.g., SETI@home), file sharing (e.g., Napster, Gnutella), instant message (e.g., ICQ, Jabber), collaboration (e.g., Groove), cooperative web-caching (e.g., Squirrel), persistent data storage (e.g., Oceanstore) to application-level multicast (e.g., Scribe), etc.

2 Challenges of P2P Networks

The topology and resource discovery are the two essential elements of P2P networks. Generally, P2P networks exhibit many common characteristics, such as large-scale, dynamic, geographical distribution and heterogeneity, etc. These characteristics make resource discover in P2P networks a challenging problem.

Many P2P networks are large-scale. For example, the registered users of Kazaa have reached 150 million in 2003. Measurement from CAIDA has also shown that the

* This work was supported by the National Natural Science Foundation of China under the grant No. 60203016 and 69933030, and the National 863 High Technology Plan of China under the grant No. 2002AA131010.

traffic of P2P applications had accounted for more than 40% of the total traffic on the backbone network in 2002.

P2P networks are strongly dynamic. Peers may join or leave P2P networks freely due to various reasons at any time. Measurement [1] on the typical P2P networks—Napster and Gnutella—has shown that the average on-line time of a peer is no more than one hour. One important reason for dynamic is that peers are highly autonomous.

P2P networks are highly distributed. There are numerous users and resources participated in the P2P network, which are geographically distributed on the Internet.

P2P networks are often heterogeneous. One reason for the heterogeneity across peers is that different peers have different capabilities, such as computing power, storage capacity, network bandwidth and etc. Another reason for heterogeneity lies in that peers have their own wills to share resources.

Security, trust and incentive are problems that P2P networks must face. Measurement has also shown that there is a “free riding” phenomenon in Gnutella network [1]: 70% of Gnutella peers share no files and 90% of the peers answer no searches. Besides the general security problems (e.g., authentication, authorization, encryption), P2P networks should be able to keep away from the malicious peers.

In conclusion, the resource discovery mechanism in P2P networks should face the main challenges below:

- Scalability. It should scale to millions of or even more peers and resources.
- Performance. It should be highly efficient for large-scale P2P networks.
- Adaptability. It should adapt well to the dynamic Internet environment.
- Resilience. It should be fault-tolerant towards peers or links failures.
- Security and incentive. It can operate correctly and effectively in an untrustworthy environment.

3 Topology Classifications of P2P Networks

There are many classification methods for topologies of P2P networks. From the view of “degree of decentralized”, P2P networks can be divided into three categories: Centralized topology (e.g. Napster) in which there is a central server to coordinate the interaction of peers; Purely decentralized topology (e.g. Gnutella, Chord) in which all peers act as both server and client equally; Partially decentralized (e.g. FastTrack, Brocade) topology in which there exist some super-nodes or super-peers that play a more important role than others.

From the view of “coupling of topology”, P2P networks can also be divided into three categories: unstructured topology (e.g. Gnutella) that is freely formed by peers; structured topology (e.g. Chord) that is precise controlled by determined algorithm; loosely structured topology (e.g. Freenet) in which the topology is freely formed by peers but the placement of data in the P2P network is controlled. The loosely structured P2P network utilizes a policy between unstructured and structured P2P networks: the network topology is arbitrary, as it is in unstructured schemes; but the placement of content is controlled, like in structured schemes. Many research topics in the loosely structured P2P networks are somewhat similar to that in unstructured P2P networks.

For convenience, we include the loosely structured topology in unstructured topology and divide P2P topologies into four categories: centralized topology, decentralized unstructured topology, decentralized structured topology and partially decentralized topology.

3.1 Centralized Topology

The centralized topology is based on a central index server (or servers) that coordinates or schedules the resources on individual registered peers. Generally the central server maintains central directories of the resources on the peers in the P2P network and coordinates the interaction between peers. Sometimes the central server can also act as a dispatcher that assigns tasks to appropriate peers.

The centralized server only provides the directory service, and the critical functions of the system (e.g., file downloading or distributed computing) are performed by distributed individual peers. Thus these systems are still peer-to-peer systems, not pure but hybrid P2P systems. Napster, SETI@home and BitTorrent are typical centralized P2P networks.

Advantages and Disadvantages. Simplicity is an important advantage of centralized topology. As the resource discovery is performed on a central directory, it can be very flexible and efficient. However, the centralized topology may introduce the single point of failure, hotspots in the network, lawsuit and other problems. It exposes vulnerability to technique failures or malicious attacks and the P2P network might completely collapse if one or some of the servers failed for some reasons.

3.2 Decentralized Unstructured Topology

In decentralized unstructured P2P networks, there is no centralized directory. When a new peer joins in the P2P network, it connects to other peers freely (e.g., selecting some random peers as neighbors). If a peer wants to publish some resources, usually it just stores them locally. Decentralized unstructured topology is well suitable for environments composed of highly autonomous peers, in which a wide range of users that come from many different organizations share resources with each other and strangers are unwilling to perform much additional work for others. Gnutella, Freenet, Mojo Nation and Nureogrid are typical decentralized unstructured P2P networks.

Advantages and Disadvantages. Decentralized unstructured P2P networks are widely deployed and predominant on the Internet for their simplicity and usability. Such systems are fault tolerant toward peers or network failures. The power-law property [2, 3] can help to explain the stable and resilient structures of Gnutella network while random failures occurring frequently. Unstructured P2P networks adapt well to the dynamic of peers and can also support rich search, such as keyword search with regular expressions, range search, etc.

However, unstructured P2P networks can only provide loose guarantee for resource discovery. Some searches may fail even if the desired resources exist in fact

and the search efficiency cannot be guaranteed. Flooding, random walk or selective forwarding is often used for resource discovery in such networks, but current search techniques are often not very efficient.

Current Research. Performance and scalability are two important open problems on unstructured P2P networks. There are many researches about unstructured p2p networks and a large part of them are focus on improving the performance and scalability. This section briefly describes several related techniques.

Blind Search. Flooding is one example of blind search method used in unstructured P2P networks. Gnutella uses blind flooding with limited time-to-live (TTL), but flooding produces too many search messages in the network. A simple approach reducing flooding traffic is to set a low TTL on initial search messages. Expanding Ring [4] and iterative deepening [5] techniques follow this idea and they increase the flooding radius in a slow way to decrease bandwidth consumption. Random walks are also suggested to take the place of flooding in many researches [4, 6].

Search with Hints. Some researches suggested that each peer in P2P network maintains some kind of metadata that can provide “hints” to guide search direction. Adamic et al. [3] proposed algorithms utilizing local information such as the identities and connectedness of a peer’s neighbors and forward search message to high degree neighbor. Directed BFS technique and local indices [5] were proposed to forward search messages to only a subset of its neighbors according to which peers with more quality results may be reached. Routing indices [7] built summaries of content that is reachable via each neighbor of the peer in different topic. Cohen et al. [8] exploited associations inherent in human selections to steer the search process to peers that are more likely to have an answer to the query. These techniques are different in the content of hints and the policy to forward message, and thus lead to different performance and characteristics.

Replication and Caching. Replication and caching are two important methods to improve search performance. In Freenet, data are proactively replicated at each peer in the path where the search message passes. Sripandikulchai et al. [9] and Markatos et al. [10] studied the characteristic of search messages and proposed some query cache policies to reduce message cost in Gnutella-style P2P networks. Cohen et al. [11] proposed and analyzed three replication strategies for blind search (uniform strategy, proportional strategy, and square-root strategy), and proved that the square-root strategy can minimize the search size.

Topology Construction and Optimization. Some researches design distributed algorithms to construct and maintain unstructured topologies with good connectivity properties for search. Raghavan et al. [12] suggested building a low diameter P2P network with high connectivity; however he didn’t discuss how to find desired data in such a system. Sripandikulchai et al. [13] exploited the interest-based locality princi-

ple, and built interest-based shortcuts among peers to improve the search performance.

Many researches have focused on constructing power-law or small-world P2P networks. Phenix [14] created a p2p network whose degree distribution follows a power-law, while its implementation is fully distributed. Zhang et al. [15] proposed an enhanced clustering cache replacement scheme which forces the routing tables to resemble neighbor relationships in a small-world network and thus improves the hit ratio of the search cache dramatically. Many other researches focused on mapping the P2P overlay efficiently to the underlying Internet network topology.

3.3 Decentralized Structured Topology

In Decentralized structured P2P networks, there are no central directories but tight control over P2P network topology. There is close coupling between the network topology and resource location information. Resource (or its metadata) is placed not on local or random peers but on specified peers by certain determined algorithms. The core component of many structured P2P networks is the distributed hash table (DHT) scheme [16, 17] that uses a hash table-like interface to publish and lookup data objects. The topology and resource discovery in P2P networks are determined by the DHT scheme.

In DHT schemes, each data object is hashed into a namespace and assigned a uniform identifier key by some public hash function. Each peer takes charge of a small part of the namespace and is also assigned a uniform peerID. In general, the data object with key O is stored on certain peer whose peerID has some mapping relationship to key O . When peers join or depart, the responsibility is re-assigned among the peers to maintain the hash table structure. Each peer also has a “forwarding table” which maintains a small number of other peers (“neighbors”) to guide routing in the P2P network. Chord, CAN, Tapestry and Pastry [17] are the most well-known DHT schemes.

Advantages and Disadvantages. Decentralized structured P2P networks and DHT schemes have attracted much attention in academic research for their desirable characteristics, such as scalability, robustness, self-management, and generality. Structured P2P networks have strong guarantees for resource discovery, i.e., the resource in such networks can be found as long as it exists and the lookup efficiency can be guaranteed. Any existing resources can be located within pre-determined hops. Thus decentralized structured topology is well suitable for environments that require a strong guarantee for resource discover, such as persistent storage system (e.g., Oceanstore). DHT schemes provide a general-purpose interface for location independent naming on which many kinds of applications can be built. For example, the DHT scheme Pastry has been used in archival storage systems (e.g., PAST), co-operative web cache (e.g., Squirrel), content distribution system (e.g., SplitStream), and etc.

However, the maintenance of DHT schemes is somewhat complex and the P2P network may churn when there is an extreme changing population of peers. DHT schemes are designed for exact-match search and can only support search by object

identifier currently. Despite these problems, DHT schemes are still very valuable research topic with a bright future for various applications.

Current Research. DHT schemes have been extensively studied these days. State-efficiency tradeoff, load balance, resilience, incorporating geography, flexible search, security and heterogeneity [16] are the main research topics. For the space limit, this section only focuses on the state-efficiency tradeoff topic.

Two important measures of DHT schemes are degree, the size of routing table to be maintained on each peer; and diameter, the number of hops a query needs to travel in the worst case. In many existing DHT schemes, such as Chord, Tapestry, and Pastry, both the degree and the diameter tends to $O(\log N)$ where N is the total number of peers in the network, while in CAN the degree and the diameter are $O(d)$ and $O(dN^{1/d})$ respectively.

An open problem posed in [16] is whether there exists a DHT scheme with $O(1)$ degree and $O(\log N)$ diameter. Recent work on Koorde, D2B, Viceroy, Fission [19] has shown that there are DHT schemes to achieve $O(\log N)$ diameter with $O(1)$ degree. D2B and Viceroy are DHT schemes to achieve expected constant degree and expected $O(\log N)$ diameter. The constant degree and $O(\log N)$ diameter of them are achieved not with certainty but “with high probability”

Xu et al. [18] systematically studied the degree-diameter tradeoff of DHT schemes and clarified the role that congestion-free plays in the degree-diameter tradeoff. A conjecture posed in [18] is that “when the network is required to be c -congestion-free for some constant c , $\Omega(\log N)$ and $\Omega(N^{1/d})$ are the asymptotic lower bounds for the diameter when the degree is no more than $O(\log N)$ and d , respectively”. The conjecture is true for a category of DHT algorithm known as uniform [18], but it is negative for general DHT schemes. For example, FissionE [20] is a $(1+o(1))$ -congestion-free DHT scheme which can achieve $O(\log_2 N)$ diameter with constant degree.

DHT schemes with constant diameter have also been proposed. For example, Ke-lips [21] is a DHT scheme with $O(\sqrt{N})$ degree and it suffices to resolve lookups with $O(1)$ time and message complexity.

3.4 Partially Decentralized Topology

Partially decentralized topology combines elements of both centralized topology and decentralized topology. There are some super-peers that own more powerful capability (processing power, storage capacity or bandwidth, etc.) than normal peers. Each super-peer acts as a centralized resource for a fraction of normal peers, and keeps the indices over the data on them. A purely decentralized P2P network is formed among the super-peers. The super-peers perform searches on behalf of the normal peers that it is responsible for. If a normal peer p wants to discover some resource, it first submits the search request to its super-peer S , and then the search request is processed among super-peers. The super-peer S acquires the search results and returns the results to peer p . But all peers are equal in terms of files downloading.

Partially decentralized topology can be two-layer or multiple-layer, i.e., there may be super-peers of super-peers in different layers. Typical partially decentralized P2P networks are FastTrack and Brocade, and the latest Gnutella also adopts the topology.

Advantages and Disadvantages. Partially decentralized topology has the potential to combine the efficiency of the centralized topology with resilience, scalability and load balance of decentralized topology. Because super-peers act as centralized servers for normal peers, the search requests might be processed more efficiently than that in decentralized topology. Furthermore, there are relatively more super-peers in partially decentralized topology, thus the problems (such as bottleneck, single point of failure, and etc.) faced in centralized topology might be avoided. Partially decentralized topology can also take advantage of the heterogeneity across peers to improve the performance.

However, partially decentralized topology might also face the similar problems in both centralized topology and decentralized topology. Super-peers play important role in the network and the failures of a few super-peers near the top of the hierarchy might have serious impact on the whole system. The super-peers also form a purely decentralized topology and face the similar problems of them.

Current Research. Despite the partially decentralized topology has been adopted in many real systems, such as KaZaa, Morpheus, the research of it is relatively few. Actually, there are many problems should be solved. For instance [22], how the super-peers are selected? How many clients should a superpeer take charge of to maximize the efficiency? How should super-peers connect to each other? How can super-peers be made more reliable? Yang et al. [22] studied fundamental characteristics and performance tradeoffs of super-peer networks in detail, and presented some practical guidelines and a general procedure for the design of an efficient super-peer network. Xu et al. [23] proposed two approaches for constructing an auxiliary expressway network to take advantage of the inherent heterogeneity of peers to speed up routing.

4 Conclusions

The P2P overlay network has become a hot topic in academic research and industry. Different topologies of P2P networks are fit for various environments and much work has been done on them. Many peer-to-peer networks have been deployed on the Internet these days, and some of them have become the most popular Internet applications. But some open problems (e.g., scalability, performance, robust, incentive) are still on the way for the success of many P2P overlay networks.

References

1. S. Saroiu, P. Krishna Gummadi, and Steven D. Gribble: A measurement study of peer-to-peer file sharing systems. Proc. of MMCN'02, (2002)
2. M. Jovanovic, F. S. Annexstein, and K.A. Berman: Modeling peer-to-peer network topologies through small-World models and power laws. In TELFOR, Belgrade, Yugoslavia, November (2001)
3. L. A. Adamic, B. Huberman, R. Lukose, and A. Puniyani: Search in power law networks. Phys. Rev. E, Vol. 64, (2001) 46135–46143

4. Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker: Search and replication in unstructured peer-to-peer networks. Proc. of ICS'02, New York, USA (2002)
5. B. Yang and H. Garcia-Molina: Efficient search in peer-to-peer networks. Proc. of ICDCS'2002, Vienna, Austria (2002)
6. Christos Gkantsidis, Milena Mihail, Amin Saberi: Random walks in peer-to-peer networks. Proc. of Infocom'2004 (2004)
7. A. Crespo and H. Garcia-Molina: Routing indices for peer-to-peer systems. Proc. of ICDCS'2002, Vienna, Austria (2002)
8. Edith Cohen: Associative search in peer to peer networks: harnessing latent semantics. Proc. of Infocom'2003, (2003)
9. K. Sripanidkulchai: The popularity of Gnutella queries and its implications on scalability. White paper, Carnegie Mellon Univ., available online at <http://www.cs.cmu.edu/~kunwadee/research/p2p/gnutella.html> , (2001)
10. Evangelos P. Markatos: Tracing a large-scale Peer to Peer System: an hour in the life of Gnutella. Proc. of CCGrid'2002 (2002)
11. Edith Cohen, Scott Shenker: Replication strategies in unstructured peer-to-peer networks. Proc. of ACM Sigcomm'2002 (2002)
12. G. Pandurangan, P. Raghavan, and E. Upfal: Building low diameter peer-to-peer networks. Proc. of FOCS'2001 (2001)
13. K. Sripanidkulchai, Bruce Maggs and Hui Zhang: Efficient content location using interest-based locality in peer-to-peer systems. Proc. of Infocom'2003 (2003)
14. R. H. Wouhaybi and A. T. Campbell: Phenix: supporting resilient low-diameter peer-to-peer topologies. Proc. of Infocom'2004, (2004)
15. Hui Zhang, Ashish Goel, Ramesh Govindan: Using the small-world model to improve freenet performance. Proc. of Infocom'2002, (2002)
16. S. Ratnasamy, S. Shenker, and I. Stoica: Routing algorithms for DHTs: some open questions. Proc. of 1st Workshop on peer-to-peer Systems (IPTPS'02), (2002)
17. H. Balakrishnan, M. F. Kaashoek, D. Karger, *et al.*: Looking up data in p2p systems. Communications of the ACM, 46(2), (2003) 43-48
18. Jun Xu, Abhishek Kumar, Xingxing Yu: On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. IEEE JSAC, No.1 (2004)
19. Dongsheng Li, Xinxin Fang, Yijie WANG, *et al.*: A Scalable peer-to-peer network with constant degree. Proc. of APPT'2003, LNCS2834, (2003) 414-425
20. Dongsheng Li, Xicheng Lu, Jinshu Su. Graph-theoretic analysis of Kautz topology and DHT schemes. Proc. of IFIP NPC'04, (2004)
21. Indranil Gupta, Ken Birman, Prakash Linga, *et al.*: Kelips: Building an efficient and stable P2P DHT through increased memory and background overhead. Proc. of IPTPS'03, (2003)
22. Beverly Yang and Hector Garcia-Molina: Designing a super-peer network. Proc. of ICDE'2003, (2003)
23. Zhichen Xu, Mallik Mahalingam, and Magnus Karlsson: Tuning Heterogeneity into an advantage in overlay routing. Proc. of IEEE Infocom'2003, (2003)

An Implementation of Semi-synchronous Remote Mirroring System for SANs

Rui Yan, Ji-wu Shu, and Dong-chan Wen

Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Abstract. Remote mirroring is often used as part of disaster recovery solutions. Synchronous remote mirroring incurs steep costs in both write latency and network bandwidth to support the mirroring, while asynchronous mirroring does not ensure the consistency of remote data. In this paper, we designed and implemented a storage-based semi-synchronous remote mirroring system for SANs. By using a log policy for the active remote write commands, this approach allows a limited number of write I/O operations to proceed before waiting for acknowledgment of receipt from the remote site, which significantly reduces write latency. This implementation also provides a consistent copy in a remote site to meet the demand for disaster recovery, because the order of commands is guaranteed. Furthermore, it can be applied to the working condition of long distance mirroring with high network latency. The testing results show that with the same request size and network latency, our semi-synchronous remote mirroring reduces average write command response time by 14-20% compared with synchronous remote mirroring.

1 Introduction

Remote mirroring ensures that all data written to a primary site are also written to a remote secondary site to support disaster recoverability. Synchronous remote mirroring is often used as part of disaster recovery solutions, such as IBM's Peer-to-Peer Remote Copy (PPRC)[1], the synchronous mode of EMC's Symmetrix Remote Data Facility(SRDF)[2] and Hitachi's Remote Copy[3]. Synchronous solutions ensure that all modifications are transferred to the remote site prior to the acknowledgement of each write to the host. Synchronous mirroring guarantees the local copies are consistent with the copies of the data at the remote site and also guarantees that the data at the remote site are as up-to-date as possible. The drawback of synchronous remote mirroring is that it adds latency to I/O write operations and requires a dedicated high-speed connection to the remote site. Furthermore, longer distances can bloat response time to unacceptable levels [1].

Asynchronous remote mirroring is also used as part of disaster recovery solutions, such as IBM's Peer-to-Peer Remote Copy asynchronous extended distance mode (PPRC XD)[1] and NetApp's SnapMirror[4]. Asynchronous solutions acknowledge a write request and allow the application executing the write to proceed prior to the modifications being sent to the remote site. The batches of

updates are periodically sent to the remote site asynchronously. Asynchronous mirroring can significantly reduce the write latency, and a lower-bandwidth connection between the local and remote copies can be used because the transfer of data is delayed. However, asynchronous mirroring does not ensure the consistency of the remote data. If the write commands arrive at the remote site out of order, the remote copy of the data may appear corrupted to an application trying to use the data after a disaster. Furthermore, asynchronous remote mirroring solutions may result in a large amount of data loss in the event of a disaster.

Semi-synchronous mirroring can be considered as a blend of synchronous and asynchronous mirroring. In semi-synchronous mirroring, write commands are sent to both local and remote storage nodes at the same time, and the application host is notified of a completed I/O when the local write is completed. Semi-synchronous mirroring is a more suitable solution, which can reduce the write latency and guarantee the consistency and currency of the remote copies. However, in current approaches such as the semi-synchronous mode of EMC's Symmetrix Remote Data Facility(SRDF)[5], a subsequent write I/O will be delayed until the completion of the preceding remote write command, while it may bring on a limited reduction of write latency and lower line utilization.

In this paper, we describe the design and implementation of an storage-based semi-synchronous remote mirroring system for the Tsinghua Mass Storage Network System (TH-MSNS)[6][7], which is an implementation of the FC-SAN. By using a log policy of the active remote write commands, this approach allows a limited number of write I/O operations to proceed before waiting for acknowledgment of receipt from the remote site, which significantly reduces write command latency. This implementation also provides a consistent copy on a remote site to meet the demand for disaster recovery, because commands arrive at the remote site in order. Furthermore, it can maintain good performance when the mirroring distance is long. In this paper, we first introduce the TH-MSNS and its remote mirroring architecture. Secondly, we describe the details of design and implementation of semi-synchronous mirroring. Finally, we discuss the testing result, which show that our semi-synchronous remote mirroring system does significantly reduce the average write command response time.

2 The Remote Mirroring Architecture for the TH-MSNS

2.1 A Brief Introduction to the TH-MSNS

The TH-MSNS is an implementation of an FC-SAN. In the TH-MSNS, the storage nodes provide storage services. A storage node consists of a general-purpose server, SCSI disk arrays, and fibre channel adapters, and it has a software module named the SCSI target simulator [8][9] running on it. By using the SCSI target simulator to control the I/O processes to access the disk arrays, the SCSI disk arrays attached to the storage node can be mapped to the host as its own local disks, on which the host OS can create file systems and databases directly. Therefore, the TH-MSNS can realize the same basic functionalities as the FC disk arrays with general SCSI disk arrays in the SAN environment. Because of

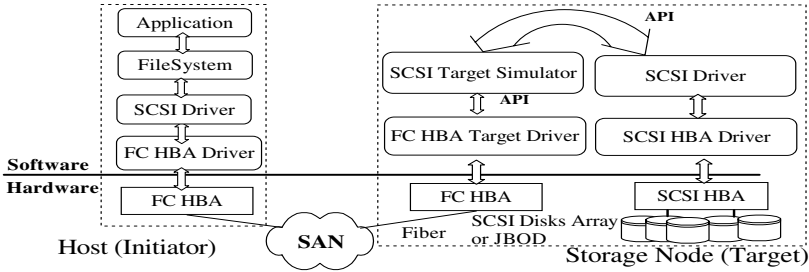


Fig. 1. The I/O path of the TH-MSNS

this, it is inexpensive, highly scalable and can achieve considerably high performance [7]. Figure 1 shows the I/O path of the TH-MSNS.

2.2 The Architecture of Remote Mirroring

We added a remote storage node to the above SAN system which has the same structure and configuration as a local storage node, and connected the two nodes with FC HBA adapters. The remote storage node's disks can be regarded as the local storage node's own disks. Therefore, the local and remote storage nodes can constitute a mirrored pair, and then the data can be mirrored from the local nodes to the remote nodes. The SCSI target simulator on the local storage node receives the SCSI commands from the server hosts, duplicates each write command into a pair of write commands for the mirrored disks, enqueues them into different request queues, and finally prompts the lower SCSI driver to process them. Hence, the local write commands are sent to the local disk, and the remote write commands are sent to the remote disk provided by the remote storage node over Fibre Channel. The remote storage node receives the remote write commands sent by the local storage node, processes and acknowledges them.

Furthermore, this remote mirroring architecture can try different linking modes to fit different distances between the two sites. With extended fabric features of the switch and Dense Wave Division Multiplexing (DWDM) technology, a remote storage node can span up to 100 km over a Metropolitan Area Network (MAN), which can significantly increase the level of disaster protection. If the distance reaches the level of a Wide Area Network (WAN), the FCIP protocol should be used, which encapsulates the Fibre Channel frames within TCP/IP packets and enables FC frames to be sent over standard TCP/IP WANs. Therefore, the remote storage node can be placed thousands of kilometers away from the local storage node, and this ensures the highest level of disaster protection. Figure 2 shows the extended architecture of remote mirroring for the TH-MSNS based IP network. In addition, the details of the design and implementation of this remote mirroring architecture have been introduced in references 10 and 11.

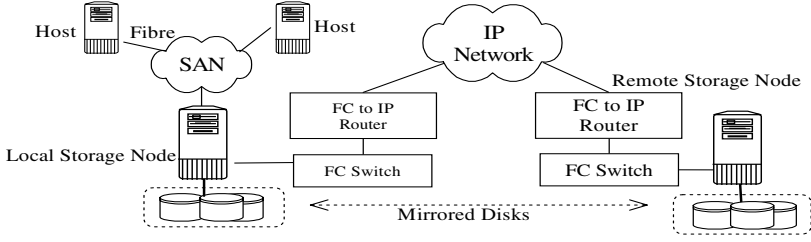


Fig. 2. The architecture of remote mirroring for the TH-MSNS based IP network

3 Design and Implementation of an Semi-synchronous Remote Mirroring System

3.1 The Semi-synchronous Write Protocol

Semi-synchronous mirroring uses a semi-synchronous write protocol. In Semi-synchronous mirroring, write commands are sent to both local and remote storage nodes at the same time, the host is notified of a completed I/O when the local write is completed, and then the remote storage node acknowledges the remote write commands when the remote write command is completed. Figure 3 shows an illustration of the semi-synchronous sequence.

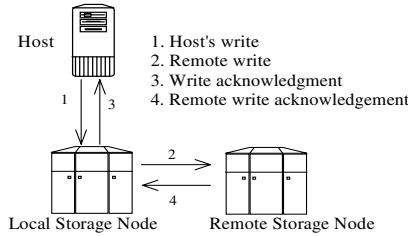


Fig. 3. Semi-synchronous write protocol

Semi-synchronous mirroring can reduce the write latency and guarantee the remote copies' consistency and currency as well. However, in current approaches such as the semi-synchronous mode of EMC's Symmetrix Remote Data Facility(SRDF)[7], a subsequent write I/O operation will be delayed until the completion of the preceding remote write command, while it may bring on limited reduction of write latency and lower line utilization. Furthermore, longer distances and higher network latency can bloat response time to unacceptable levels in some applications, which require a fast response time, such as Online Transaction Processing (OLTP). In order to improve command response time and line utilization, especially in the conditions of lower-bandwidth connections such as IP networks, we designed and implement an semi-synchronous remote mirroring system for TH-MSNS.

3.2 Semi-synchronous Remote Mirroring

In the semi-synchronous remote mirroring process we propose, when a write command is received from the application host, the local storage node converts it into a pair for write commands to the mirrored disks. Before dispatching of the local and remote write command, the corresponding information of the remote write is recorded into a command log, which may appear as a specific data buffer in the main memory. The application host is informed of an I/O completion when the local write is command completed, while the data buffer of this command is not released for the moment. The data buffer and the corresponding records in the command log will not be released until the acknowledgement of the remote write command arrives. Because the command log records all remote write commands not acknowledged, the semi-synchronous mirroring system can allow a limited number of write I/O operations to proceed before waiting for acknowledgment from the remote site. Therefore, a subsequent write I/O does not need to wait until the completion of the previous remote write command, and can be dispatched without interruption. This approach significantly reduces the latency of write commands and considerably improves the line utilization. Furthermore, write commands can arrive at the remote site in order, which guarantees the consistency of the remote copy.

3.3 Command Log Policy and Re-synchronization

The command log is a special data buffer in the memory, which records the information of the remote write commands without acknowledged them. It is organized as a queue to contain the corresponding information of the remote write command, such as SCSI CDB, data buffer pointer, data buffer length and the destination of the command etc. Each remote write command puts its information into the command log before being transferred, and each record will be cleared after the remote write command is acknowledged. The maximum length of the command log is the maximum number of write I/O operations that can be allowed to proceed before acknowledgement. If the length of the command log reaches the maximum value, all subsequent read/write commands will be blocked until the length of the command log decreases to less than the maximum length again. Figure 4 shows the structure of the command log.



Fig. 4. The main structure of the command log

Since remote mirroring does not depend on a real connection between the local and remote site, a break of the connection will cause additional update

information to be queued in the command log, and the write commands will only be sent to the local storage device. If the command log becomes too large, or data buffer is full, the log will be cleared and the remote storage will be marked as requiring a full re-synchronization when a connection becomes available again.

3.4 The Processing of Write Commands and Its Petri Net Model

In the mirroring architecture we designed, the SCSI target simulator on the local storage node receives the write commands and the data from the FC target driver, then allocates the data buffers for these commands and queues them. After the SCSI target simulator receives the pending data of one corresponding command and fills up the data buffer, the command is ready to be dispatched. The mirror sub-modules of the SCSI target simulator converts each write command into a pair of write commands for mirrored disks and records the remote write command in the commands log, and finally prompts the SCSI driver to process them. Hence, the local write commands are sent to the local disk, and the remote write commands are sent to the remote disk provided by the remote storage node over Fibre Channel. The host is notified of a completed I/O operation when the local write command is completed. After the remote write command is completed, the data buffer and its corresponding records in the command log will be released. If the length of the command log reaches the maximum, the SCSI target simulator stops to dispatch the queued commands until the length is less than the maximum length again. Figure 5 shows a Petri Net model of the write command processing flow in semi-synchronous remote mirroring.

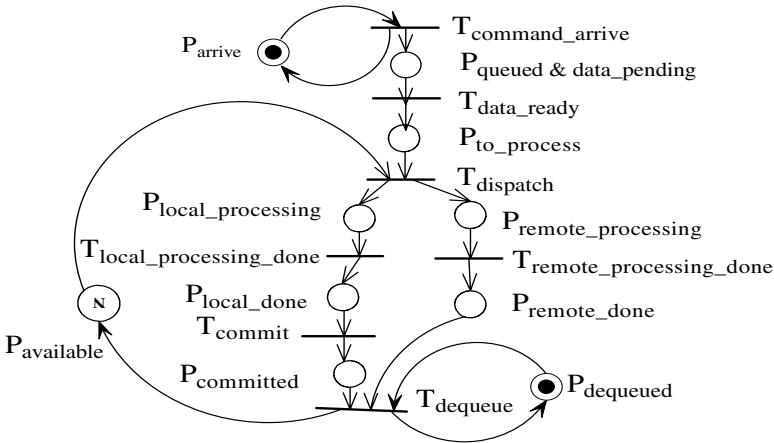


Fig. 5. Petri Net model of the write commands Process

4 Performance Evaluation

Tests were designed and performed to evaluate and analyze the performance of the semi-synchronous remote mirroring system. Because read commands are

Table 1. Test configuration of hosts and storage node

Host		Storage Node	
CPU	Intel Xeon 2.4GHz × 1	CPU	Intel Xeon 2.4GHz × 1
Memory	1G	Memory	1G
OS	Linux (kernel: 2.4.18)	OS	Linux (kernel:2.4.18)
FC HBA	Emulex LP982(2Gb/s)	FC HBA	Emulex LP982(Initiator, 2Gb/s) Qlogic 2300(Target, 2Gb/s)
		RAID Card	Adaptec RAID Card 2110S
		SCSI Disk	Seagate Cheetah(10KRPM) 73GB × 7, JBOD

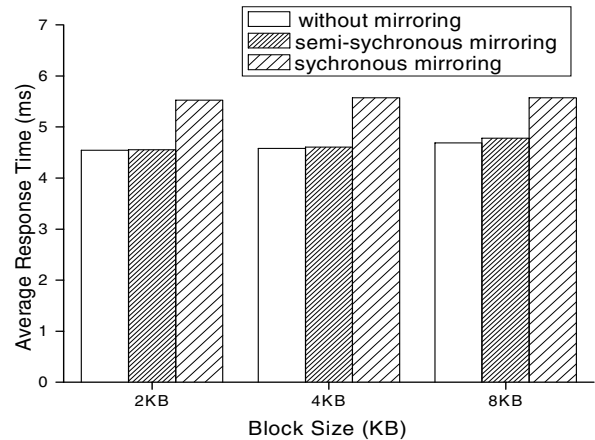


Fig. 6. Average response time with 3ms remote network delay

executed locally in the process of mirroring, our tests only focused on the write commands. Table 1 shows configurations of the hosts and storage nodes.

The average response time of commands is a very important factor to evaluate the performance. For example, users of Online Transaction Processing (OLTP) applications must wait for each commit before proceeding. In this test, a host issues write commands with different data block sizes to its 'network' disk, which is provided by the local storage node. The goal is to compare the average response time of the commands with synchronous and semi-synchronous. In order to evaluate and analyze the system's performance in conditions with long distances and high-latency connections between the local and the remote site, we introduced some software delays in the processing procedure of the commands on the remote storage node. The IOMeter [12] benchmarking kit was used, and the open mode of physical disks was O_DIRECT. The host issued random write commands with block sizes ranging from 2KB to 8 KB. The length of the command log was 30. Figure 6 and figure 7 show the test results.

The results show that with the same request size and network latency, our semi-synchronous remote mirroring reduces average write command response

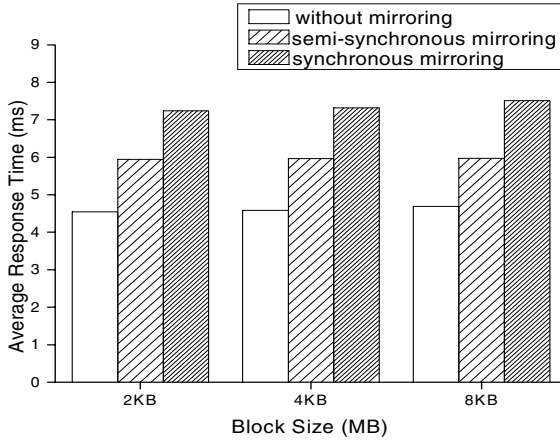


Fig. 7. Average response time with 5ms remote network delay

time by 14-20% compared with synchronous remote mirroring. Since a large number of applications and database managers have a fixed I/O request size, such as Microsoft SQL Server (8KB), the Oracle RDBMS (8KB), or Microsoft Exchange Server (4KB), the test conditions were very similar to a real environment. When the network latency is low (3ms), semi-synchronous mirroring adds very low latency to write I/O operations compared with no mirroring.

When the network latency increases (5ms), the write latency brought by semi-synchronous mirroring also increases, but is still lower compared with synchronous mirroring.

5 Conclusion

This paper describes the design and implementation of a semi-synchronous remote mirroring for the TH-MSNS. In order to eliminate the drawbacks of traditional semi-synchronous remote mirroring implementations, we proposed a log policy for active remote write commands. Compared with other systems, the new system has the following advantages: 1. It significantly reduces write latency and efficiently improves the line utilization. 2. It also provides a consistent copy on a remote site to meet the demand for disaster recovery. 3. It can be applied to conditions with long mirroring distances and high network latency.

Acknowledgements

The study is supported by National High-Tech Research and Development Plan of China under Grant No. 2001AA111110.

References

1. A.C.Azagury, M.E.Factor, W.F.Micka. Advanced Functions for Storage Subsystems: Supporting Continuous Availability, IBM SYSTEM Journal VOL 42, 2003.
2. Using EMC SnapView and MirrorView for Remote Backup, Engineering White Paper, EMC Corporation. (April 2002)
3. Software Solutions Guide for Enterprise Storage, Hitachi Data Systems Corporation. (December 2000)
4. Patterson, R.H., et al. SnapMirror: File-System-Based Asynchronous Mirroring for Disaster Recovery. In First USENIX conference on File and Storage Technologies. 2002. Monterey, CA, USA
5. Neil E.ERVIN. Managing Extended Distance EMC SRDF Semi-Synchronous Remote Copy. Technical report. (August 1999)
6. Shu Ji-wu, et al. A Highly Efficient FC-SAN Based On Load Stream. Xingming Zhou, Stefan Jahnichen, Ming Xu, and Jiannong Cao eds., The Fifth International Workshop on Advanced Parallel Processing Technologies, LECTURE NOTES IN COMPUTER SCIENCE 2834, pp.31-40, 2003
7. Technical Report: Design and Implementation of the TH-MSNS (In Chinese), Computer Science Department, Tsinghua University, P.R. China, 2003.
8. Ashish Palekar, Narendran Ganapathy. Design and Implement of A LINUX SCSI Target for Storage Area Networks. Proceedings of the 5th Annual Linux Showcase & Conference, 2001. Oakland, USA.
9. Jae-Chang Namgoong, Chan-Ik Park. Design and Implement of a Fibre Channel Network Driver for SAN-Attached RAID Controllers. IEEE Parallel and Distributed Systems, 2001.
10. SHU Ji-wu, YAN Rui, WEN Dong Chan, ZHENG Wei-min. An Implementation of Storage-Based Synchronous Remote Mirroring for SANs, IFIP International Conference on Network and Parallel Computing 2004 (Accepted).
11. YAO Jun, SHU Ji-wu, ZHENG Wei-min, A Distributed Storage Cluster Design for Remote Mirroring based on Storage Area Network, IEEE Transactions on computers (Submitted), also in <http://storage.cs.tsinghua.edu.cn>
12. Jerry Sievert. Iometer: The I/O Performance Analysis Tool for Servers. <http://www.intel.com/design/servers/devtools/iometer/index.htm>

A Security Scheme for United Storage Network

Yihui Luo^{1,2}, Changsheng Xie¹, and Chengfeng Zhang¹

¹ National Storage System Laboratory, School of Computer Science
Huazhong University of Science and Technology, Wuhan, Hubei, P.R. China, 430074
lyh@huhu.edu.cn

² School of mathematics and Computer Science, Hubei University
Wuhan, Hubei, P.R. China, 430062

Abstract. USN realizes the integration of SAN and NAS with IP network, but it brings new security consideration such as user authorization, data privacy and integrity. A USN model based on the third party transfer protocol is suggested to realize the security scheme. This security scheme has the following characteristics: A key distribution scheme is used to create credentials for users in order to reduce authorization server performance penalty; Using HMAC authenticates users requests so as to minimize computation overhead; Performing encryption/decryption of data at clients and storing data checksums on the storage will minimize the storage performance penalty; The lockbox is used to integrate keys in order to minimize the sum of keys need managed by authorization server. Experiments show that it takes less than 10% performance overhead to realize the security scheme for USN comparing the baseline USN.

1 Introduction

NAS and SAN are the leading network storage. NAS links storage devices directly to user network to provide file service for users, which provides apparently file management and file sharing in heterogeneous environment [3]. SAN links servers and storage devices with FC and provides users with data block service, which provides high reliability and scalability [1]. NAS and SAN, having disparate characteristics, are used for different applications. However the growth of application requires NAS and SAN to exist at the same time, a new storage network scheme called USN (United Storage Network) is used to satisfy this demand [4]. USN puts many storage devices into a single storage space with IP network, which provides simultaneously users with file service and data block service.

In SAN scheme, the storage network as a private network is different from the user network. Users can access storage devices only through servers, so storage security can be realized though servers' zoning and fencing; NAS is often used in LAN which security is easy to ensure. However, in USN scheme, storage network is the same network of user and users can directly access storage devices though direct channel, so the storage devices will be exposed in the user network, that leads to new security problems. Considering performance and security, we design and implement a security scheme for USN with lower performance penalty.

2 USN Model

The USN model is described as Figure-1. Its includes NAS devices and SCSI disk, the NAS devices are linked to IP network directly, the SCSI disks link to IP network through IP disk controller [4]. All those disks are integrated into a single storage space though the metadata server. Clients can access disks through direct channel after getting metadata from the metadata server because user network and storage network belong to the same IP network. Clients access storage devices according to the third party transfer protocol. The third party transfer protocol is described as the followings: Firstly, clients send I/O requests to the metadata server for metadata when they need to access the storage devices; Secondly, the metadata server authenticates clients and returns clients with metadata which contains storage devices IP addresses, file information (when it is file I/O), data blocks location on the storage devices and clients accessing rights; Thirdly, clients request the storage devices with metadata for block I/O(as it is SCSI disks) or file I/O(as its NAS devices); Lastly, the storage devices authenticate I/O request and permit the clients accessing the storage devices.

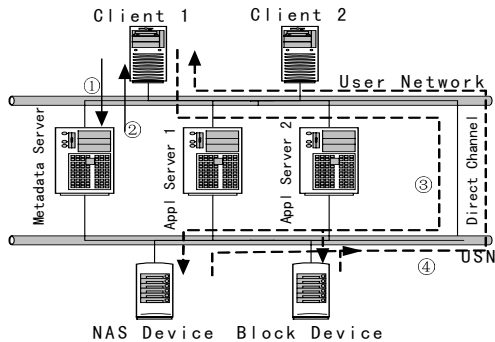


Fig. 1. USN Architecture

The USN model has some advantages by using the third party transfer protocol. Managing centrally storage devices makes clients share the storage space and improves the storage devices efficiency, even provides the clients with a storage space larger than the total space of all storage devices. In addition, using third party transfer protocol can improve the I/O performance.

3 USN Security Analyses

In USN scheme, storage devices are directly linked to user network and users can directly access storage devices, which will lead to new security concerns: since it is the client, not the server, that initiates I/O requests, storage devices can no longer trust every request received. In addition, placing storage devices as the first class network entities exposes them to the similar types of attacks that only the servers faced in SAN: malicious parties forging messages or tampering with message con-

tents, replaying or recording messages, spoofing user's identity or denying service of valid requests. All those make data stored on storage insecure.

In order to realize USN security, we use the credential distribution scheme to authenticate I/O request, in which the metadata server is used as an authorization server authenticating clients and distributing credentials of data objects. So the USN security model includes three parties: clients, storage devices and an authorization server, as depicted by Figure-2.

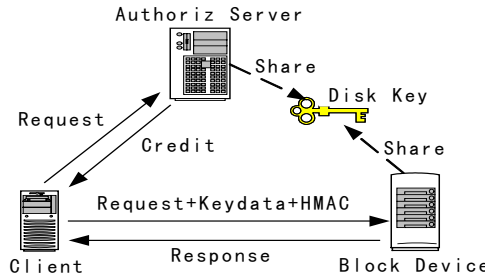


Fig. 2. USN Security Model

Storage devices are envisioned as trusted entities because data integrity is preserved on storage devices and they send the right data back to the client upon validating an authorized request. We suppose that they have lower computation ability because storage devices are usually controlled by microprocessors.

Clients are not trusted. Some of them may in fact be written by the adversary, and others may run on machines that are compromised. Since computer performance becomes more and more high, clients are considered to have high performance.

Authorization server is highly trusted, which runs on a secure machine that is capable of storing long-lived keys, it truthfully determines the access rights by distributing credentials.

Communication links are not secure because they are realized through IP network.

4 USN Security Scheme

4.1 Keys

The authorization server has a pair of private and public keys. The public key is informed to both disk controllers and clients as they take part in USN. Each client has a pair of private and public keys that is used to user authorization. Each disk controller only contains a unique disk key K_d that is shared by the authorization server and used to create credential.

Since disk controllers are implemented by lower performance processors, data are encrypted/decrypted at clients to minimize disk controllers' performance penalty.

Every object (as a file or a group of blocks) contains several blocks, and each block is encrypted with a symmetric key called a data block key. In order to reduce the number of keys the authorization server needs to manage, distribute and receive, we use a lockbox to hold the data block keys of the object. The lockbox refers to encrypting data block keys with another key named lockbox key K_{lockbox} . The lockbox keys are given to authenticated clients by the authorization server.

A storage device authenticates a client's I/O request with a credential. A credential contains two parts: a secret key hk_u and a secret key data KeyData. The secret key hk_u is created by hashing the KeyData with K_d , and the KeyData may include client's access rights on one or more objects. The credential is produced by authorization server and sent back to the client as it request accessing an object for the first time.

4.2 USN Security Protocols

USN security is realized through user authentication, I/O request authentication, data privacy and data integrity. A user first has to be authenticated by the authorization server before obtaining a credential from the authorization server. I/O request authentication is realized with credential authentication. In order to reduce storage devices performance penalty, data privacy is realized though data encryption at clients so that data in cryptograph are stored on storage devices and transferred over IP network. Data integrity is provided by both block checksums and HMAC. Block checksums are used to check data integrity and are stored on storage devices. HMAC is used to both ensure data integrity in transfer and authenticate the I/O request.

4.2.1 Disk Key Sharing Mechanism

The credential distribution is based on a set of disk keys that are shared by the authorization server and storage devices. In our USN security model, a disk key is created from the disk information. As a storage device first joins USN, it gets the public key of the authorization server from the management server. On subsequent I/O operating on storage devices, storage devices send disk information encrypted with the public key of the authorization server to the authorization server that decrypts the disk information with its private key. So both storage devices and the authorization server have the same disk information. Using the same arithmetic, the authorization server and storage devices create the same disk key K_d from the disk information.

4.2.2 User Authorization

When a user wishes to access USN, he has to send authorization request information (AuthMsg) to the authorization server. AuthMsg includes user's authorization request (AuthReq), request sequence number (SeqNo) and user public key K_{up} . The client assigns a unique SeqNo for each AuthReq to check whether the user request is outdated or not. In order to ensure request integrity, AuthReq concatenated with SeqNo is hashed with SHA-1 and then signed with user private key K_{us} . AuthMsg is encrypted with authorization server public key K_{ap} in order to achieve it privacy.

$$\text{AuthMsg} = \{\text{SHA-1} \{ \text{AuthReq}, \text{SeqNo} \}_{K_{us}}, K_{up}\}_{K_{ap}} \quad (1)$$

After receiving AuthMsg, the authorization server decrypts AuthMsg with its private key K_{as} , verifies the signature using K_{up} , and checks that SeqNo has not appeared before. If all of these are valid, the authorization server will authenticate user by consulting user identity database and map the user public key to the user ID. If everything succeeds, the authorization server will distribute credential to the user.

4.2.3 Credential Creation

Credential creation is based on key distribution protocol, and credentials are created with the object information and disk keys. A credential contains two parts: a key data KeyData and a secret key hk_u .

$$\text{KeyData} = \{\text{UserID}, \text{ObjectID}, \text{Metadata}, \text{ExpTime}\} \quad (2)$$

Where, ExpTime is used to denote the credential living time. The secret key hk_u is generated by hashing KeyData with a corresponding disk key K_d .

Afer creating credential, authorization server returns user with authorization response AuthRes:

$$\text{AuthRes} = \{\text{Metadata}, \text{KeyData}, hk_u, \text{SeqNo}, K_{\text{lockbox}}\}_{K_{up}} \quad (3)$$

The lockbox key is used to integrate data block keys of the object, and SeqNo is used to map AuthRes with AuthReq. AuthRes will be encrypted with user public key K_{up} .

4.2.4 Request

After decrypting authorization response from the authorization server, clients can access the storage devices with credentials. The client I/O requests are the following:

$$\text{Request} = \{M, \text{Data}, \text{CheckSum}, \text{KeyData}, \text{HMAC}\} \quad (4)$$

The request information M contains object metadata, SeqNo, lockbox. Data are divided to blocks and data blocks are encrypted with data block key K_s that are contained in the lockbox. Data checksums are computed on encrypted data blocks. HMAC is computed on M, Data and checkSum with hk_u . It is noted that there is no data and checksum if the request is a READ operation.

4.2.5 Response

After receiving I/O request, the storage device first uses K_d to hash KeyData to generate hk_u with the same arithmetic on the authorization server, and then computes HMAC as it does at the client. If the computed HMAC is equal to the received HMAC, the request is legal and the storage device responds to it. The storage devices can't compute right HMAC on storage devices if M, data, CheckSum, KeyData, or all of them are modified as they are transferring, the computed HMAC is not possible to

equal to the received one, so HMAC not only provides credential authentication, but also checks the integrity of M, Data and CheckSum. If the request is a WRITE, the storage device stores its data, checksum and lockbox; if the request is a READ, data, checksum and lockbox are read from the storage device. The response to the client is in the following form:

$$\text{Respon}=\{M, \text{Data}, \text{CheckSum}, \text{HMAC}\} \quad (5)$$

It's noticed that the KeyData is not included in the response since the client already possesses the user's credential. The HMAC is only computed on M because the integrity of the data is checked though CheckSum, which will cut down the storage device computation penalty. In particular, there are no data and checksums as it is for a WRITE.

5 Performance Test and Estimation

We have realized USN and its security scheme in lab. The USN consisted of an authorization server, a client and two IP disks. They were connected to each other with 1000 Mb/s Ethernet using a switch. The authorization server and client were all configured with P4 1500 CPU, 256MB RAM and RedHat7.1; The IP disks were Seagate ST318437LW SCSI disks that were linked to the network with IP disk controller configured with P3 730 CPU, 128MB RAM and RedHat7.1.

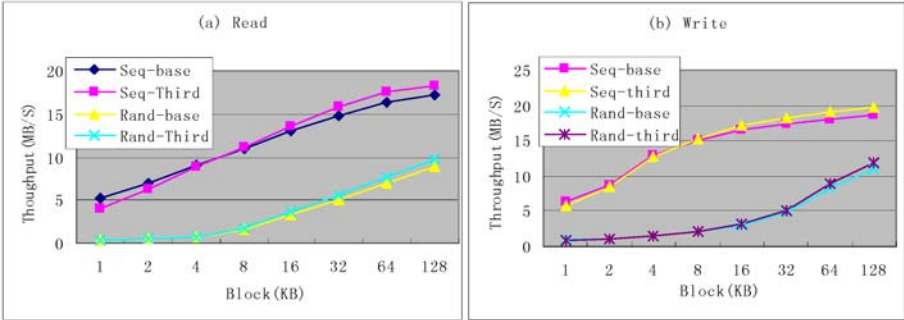


Fig. 3. Performance comparison of baseline USN and USN based the third party transfer

The USN used Intel's iSCSI-v8-intel software to realize data transfer over the network. The initiator software was running at the authorization server and the target software was running at the IP disk controller. The client was also configured with the initiator software in order to encapsulate/decapsulate data blocks. The iSCSI software has been modified to realize the third party transfer protocol. In order to test the performance penalty of the security scheme, the authorization server ran key distribution scheme to create credential as it distributed metadata to the client. The client encrypted data, computed data checksum and HMAC of request before accessing the storage. The disk controller authenticated the client's I/O request.

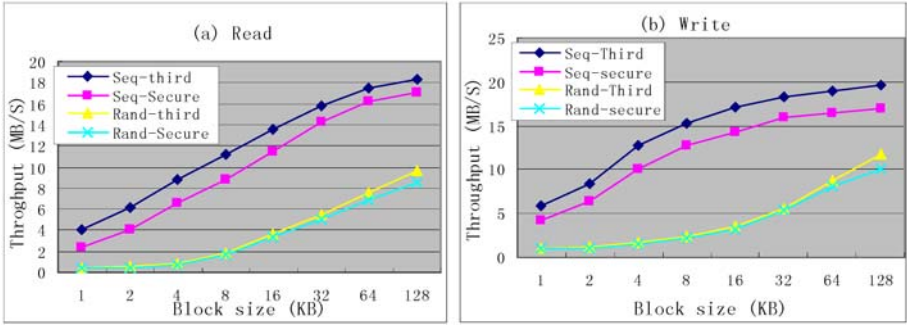


Fig. 4. Performance comparison of USN based third party transfer protocol and USN with security scheme

In lab, we used Iometer to test the I/O throughput at the client. The performance comparison of the baseline USN without security and USN based the third party transfer is described as figure-3. From figure-3, we can find that the I/O performance of the USN based the third party transfer protocol is better than that of the baseline USN with the size of data blocks grows. This is because data must be transmitted though server in the baseline USN while data is transferred directly in the other.

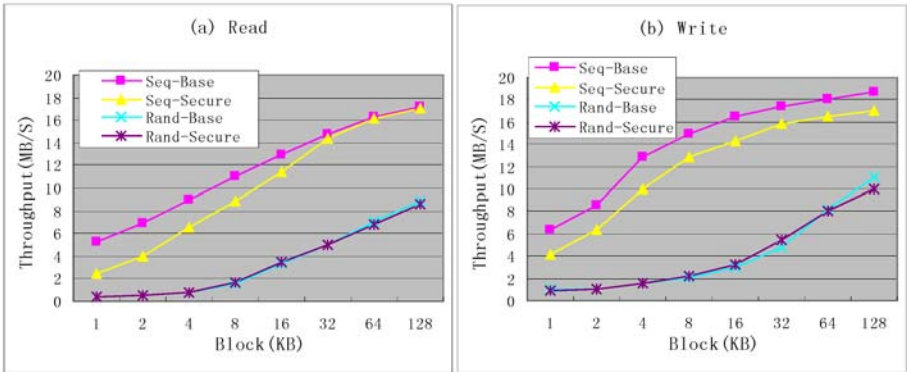


Fig. 5. Performance comparison of the baseline USN and the USN with security scheme

Figure-4 shows the performance of USN based on the third party transfer protocol with/without security. The performance of sequential accesses is much better than that of random accesses, but random accesses suffer less performance penalty for security than sequential accesses. This is because the performance penalty of security is determinate and the sequential access time is shorter than the random access time. We can also find that sequential writes with security have the worst performance penalty with block sizes increasing. As block sizes become more than 4KB, the performance penalty is between 12-25%. It is because Write must compute checksum and HMAC at client and the computation overhead is dependent on the sizes of data blocks. We can find that the performance penalty of Reads becomes small with block

sizes increasing. When block sizes are larger than 32KB, the performance penalty is less than 9%. The reason is that the disk exploits data block checksums to achieve good read performance.

Figure-5 compares the performance of the baseline USN and the USN based the third party transfer protocol with security scheme, when the size of the data blocks is larger than 32KB, the performance of sequential read, random read and write is close to that of the baseline USN, while the random write performance penalty of the security scheme is less than 10% comparing the baseline USN. This is because part of the security overhead is paid by the third party transfer protocol.

6 Conclusions

USN integrates SAN and NAS, and provides file I/O service and data block I/O service at the same time, but it also brings new security problem. We have designed and implemented a security scheme for it. The security scheme has the following characteristics: (1) A key distribution scheme is used to reduce authorization server performance penalty; (2) Using HMAC for requests authentication and performing encryption/decryption at clients to protect data privacy and integrity minimize storage devices performance penalty; (3) Storing data checksums on storage devices minimizes storage devices computation penalty, so the performance bottleneck of USN is moved from storage devices to clients; (4) Using lockbox to integrate keys can minimize the sum of keys need managed by authorization sever; (5) An experiment shows that USN security scheme requires less than 10% performance penalty comparing the baseline USN.

References

1. Kaladhar Vorugaanti and Prasenjit Sarkar: An Analysis of Three Gigabit Networking Protocols for Storage Area Networks. IEEE,2001.
2. Charles Baumert: Secuing IP Storage Networks. Cylink Corporation. October 2002,USA.
3. Garth A. Gibson and Rodney van Meter: Network Attached Storage Architecture. Communication of Theacm. November 2000.
4. WANG Zhiheny and BAI Yingcai: A New Scheme of Integrating NAS with SAN. Journal of Shanghai Jiaotong University. Vol. E-8, No.1,2003.
5. Yingwu, Zhu Yiming Hu. SNARE: A Strong Security Scheme for Network-Attached Storage. The 2nd International Symposium on Reliable Distributed Systems (SRDS'03). 2003.
6. E. Reidel, M. Kallahalla and R. Swaminathan: A framework for evaluating storage systems security. The 1st conference on File and Storage Technologies (FAST), January 2002.
7. E. Miller, D. Long, W. Freeman and B. Reed: Strong Security for Network-Attached Storage. the FAST 2002 Conference on File and Storage Technologies, January 2002.
8. B. C. Reed, E. G. Chron, R. I. C. Burns, D. E. Long: Authenticating Network-Attached Storage. *IEEE Micro* Vol. 20(1), pp. 49-57, January 2000.

STS: A Share Taper System for Storage Area Networks

Jiefeng Xu and Zheng Qin

Xi'an JiaoTong University, Shengli oilfield Information Center, China
xjf@slof.com

Abstract. Data backup is an effective way to protect data. In storage area network (SAN) environments, the traditional data backup method can not meet the needs of data backup. We proposed and implemented a multi-server backup maintenance system Share Taper System (STS). In this system, multi-servers share the tape devices through TCP/IP or Fabric. We also implemented an exclusive lock mechanism based on SCSI-3 to harmonize multi-servers' requirement for tape devices. The test results showed that the STS can effectively improve the utilization of backup systems, and is compatible for isomerous platforms. STS does not require users to update their existing backup software.

1 Introduction

A Storage Area Network (SAN)[1] is a storage architecture based on a network. The SAN separates data storage from the server, and has flexible addressing capability, high data transfer speed, high I/O performance, and high share ability. The SAN has become the most primary solution for high storage performance and reliability required by applications. At present, Fabric and Ethernet are two popular networks used in SANs.

Data is the most precious wealth for enterprises. Data backup is a traditional and effective way to protect data. With the development of massive network storage technology, especially SAN technology, traditional data backup methods can not meet the needs of data backup in network storage environments. Directly attaching tape devices to a server, traditional data backup technology can not meet the needs of isomerous platform servers. At present, FC tape devices can be shared by isomerous platform servers through a Fabric network. But they lack an exclusive mechanism and can only be used in Fabric. Traditional SCSI tape devices can not sustain isomerous platform servers, while data needing to be backed up is often distributed on isomerous platform servers, which increases the workload of data backup.

At present, the popular lock mechanism in multi-servers is the Distributed Lock Manager (DLM)[2]. DLM is implemented on servers and provides share access to storage for cluster systems in distributed environments. DLM can ensure a consistent view for servers in a cluster system. But DLM can not support isomerous platforms. In this paper, we present a lock mechanism based on devices:

Device Lock: Mutual Exclusion for Storage Area Networks[3]. It is an expanded command of SCSI-3 and provides a lock mechanism in a distributed environment.

Enterprises usually need to back up their data every week. The data could be scattered over several servers, and this requires many sets of tape drives to accomplish a full data backup. The Share Taper System (STS) can effectively share different tape devices' magnetic heads, allow synchronous backup of the data on several servers, reduce the system backup time, and increase the utilization of the backup resources. And based on STS, we designed and implemented a mechanism to share backup resources for multi-server synchronous access. This system can be built on TCP/IP (IP-SAN) or Fabric (FC-SAN) to accomplish the sharing of backup resources. It is implemented on the device driver layer and directly handles SCSI orders, so it is independent of backup software in the application layer. Thus users can continue to use their existing backup software.

The test results indicate that the STS can successfully allow sharing of backup resources, efficiently utilize backup resources, and reduce backup time. The STS is especially applicable for backup systems in SAN environments. And we also implemented an expanded SCSI-3 command lock mechanism to realize a consistent view of multi-servers and ensure the correctness of the data backup.

2 Architecture

2.1 The Construction of the Whole STS

Figure 1 shows the hardware architecture of the STS system. The SAN can be a Fabric or Ethernet network. The SCSI tape device is connected into a massive network storage system through an I/O node machine. The STS system software runs on the I/O node machine, and is shared by servers through network export in the I/O node machine.

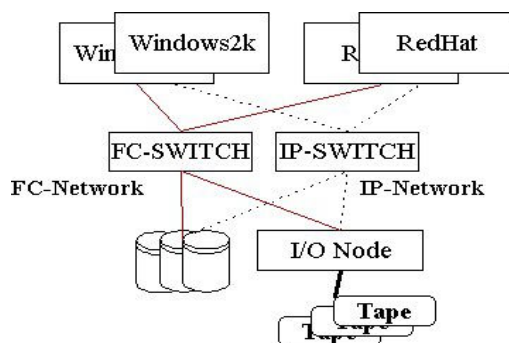


Fig. 1. The Hardware Architecture of the SAN System with STS

The SCSI target simulator in the I/O node receives the SCSI command produced by the server, and transmits it to the STS. Then the STS chooses the proper SCSI tape device to carry out the command.

In the design of the STS, we focused on the interface between the STS and the SCSI target simulator, which makes the STS compatible with existing FC-SAN systems and enables the STS to work on IP networks and form an iSCSI system. The HBAs or iSCSI driver modules that work separately as initiator and target constitute the data and command path. In the initiator, the HBA registers to the SCSI middle level as the SCSI lower level and forms the data and command path in the initiator; in the target, the HBA registers to the STS and the STS is responsible for the command's execution and data transmission. The STS and SCSI system on the target are joined together to control storage resources and execute SCSI commands, and accordingly form the entire data and command path.

2.2 Design of the Interface

The I/O node machine is connected into the SAN through the HBA. It is responsible for receiving SCSI commands and data from the initiator and then transmitting them to the STS. We defined an interface between the STS and HBA driver working as a target, which is the same in FC-SAN and IP-SAN. The interface was defined as follows: The primary interface functions that STS provides to the HBA driver are as follows.

```
Target_Scsi_Cmnd *rx_cmnd (Scsi_Target_Device *, __u64, __u64, unsigned
char *, int)
```

```
int scsi_rx_data (Target_Scsi_Cmnd *)
int scsi_target_done (Target_Scsi_Cmnd*)
int rx_task_mgmt_fn (__u64,__u64)
```

The primary interface functions that the HBA driver contributes to the STS are as follows.

```
int (* detect) (struct STT*)
int (* release) (struct STT*)
int (* xmit_response)(struct SC*)
int (* rdy_to_xfer)(struct SC*)
```

After the STS carries out a command, it prompts the xmit_response function to inform the lower driver. And the rdy_to_xfer function is used by the STS to inform the lower driver that the data buffer is ready and then the STS and the lower driver are synchronized for data transfer.

After receiving a SCSI command, the STS must control the execution of the tape devices linked into the I/O node. And this can be implemented through the sg interface of the SCSI system, or directly through the middle level of the SCSI system. If implemented through the sg interface, the sg interface becomes an interface whose kernel is provided for users to directly execute SCSI commands. It is easy for users to explore and debug, but it will increase the time which the sg module takes to handle SCSI commands, and this will effect the whole performance. So we used an interface implemented by scsi_mod in the STS. The primary interface function for executing SCSI commands is scsi_do_request.

Because all the read/write commands need some memory to store data, the STS was designed with a memory storage pool. The size of the pool can be

configured, and can be dynamically adjusted according to the requirements when working. When a new SCSI command needs memory, it can directly apply from the pool, and then directly release resources to the pool after being executed. In this way, the delay and complexity associated with applying and releasing memory can be reduced.

Therefore, the STS can be divided into four modules: the SCSI command handle module, the SCSI message handle module, the command/data receive module, and the command/data send module. And the STS needs to maintain the SCSI command queue, the SCSI message queue, the tape device information queue, and the memory storage pool. All SCSI commands and data delivered by the SAN system need to be handled by the STS, so the STS can supervise the data flux according to system time; namely STS has an interface that performed well in tests.

2.3 Working Flow of STS

The STS system consists of two queues: the SCSI command queue and the SCSI message queue. The two queues are handled in the same way, so only the command queue will be described.

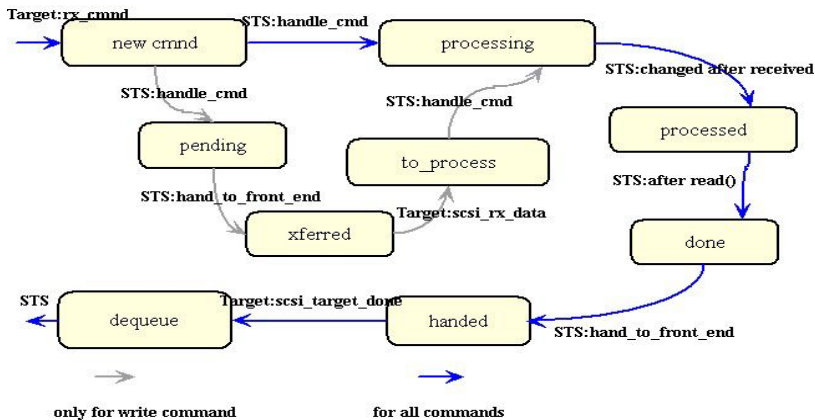


Fig. 2. The SCSI command status in the STS

A SCSI command has 8 states in the STS; they are new_cmdnd, processing, pending, xferred, to_process, done, handed and dequeue. Figure 2 illustrates the status changes of a SCSI command. Note that the write data command, which moves data from target driver to STS, has one more status than the read data command.

The status changes of the WRITE_6 command are described in detail to show how write commands are executed in the STS.

Processing of SCSI Command. As shown in figure 3, the WRITE_6 command is executed in 8 steps:

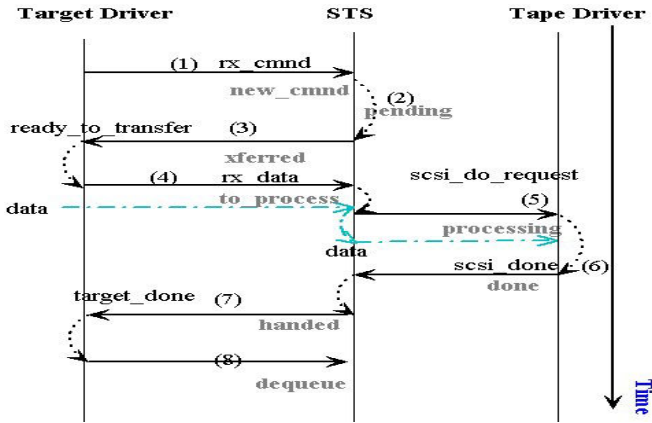


Fig. 3. Processing of the write_6 SCSI command

1. The target driver receives a new SCSI write command. It prompts the STS function `rx_cmnd` to allocate data structures for the new command. The status of the command is changed to `new_cmnd`.

2. The command-processing thread of the STS processes the write command. It allocates memory for data to be written to tape according to information in the CDB and changes the command status to `pending`.

3. The STS notifies the target driver that the memory space is ready and changes the command state to `xferred`.

4. The target driver writes data to the allocated memory space and changes the command status to `to_process`.

5. The command-processing thread prompts the `scsi_do_request` function in the SCSI mid-layer to execute the command. The command status is changed to `processing`.

6. The tape driver finishes the command, prompts the handler functions of the STS to perform the verification and changes the command status to `done`.

7. The STS notifies the target driver of the completion of the command and returns successfully. The command status is changed to `handed`.

8. The target driver finishes the processing of the command. The command status is changed to `dequeue`. The command-processing thread of the STS recycles the allocated resources and the command is over.

The process of the Read_6 command is simpler than the write command described above. It does not have the `pending`, `xferred` and `processed` states. According to SCC [4][7] in the STS system, SCSI write commands that need to be processed are commands such as `MODE_SENSE`, `MODE_SELECT`, `WRITE_6`, and `SEND_DIAGNOSTIC`. Read commands include `READ_BLOCK_LIMIT`,

READ_6, REQUEST_SENSE and RECEIVE_DIAGNOSTIC. Control commands include WRITE_FILEMARKS, SPACE, START_STOP, ERASE, REZERO_UNIT, RESERVE, RELEASE, and SET_LIMIT.

The SCSI command READ POSITION should be carefully implemented. This command reports the current position and provides information about logical objects contained in the object buffer.

3 Implementation of Device-Based Exclusive Lock

The device-based exclusive lock implementation is platform-independent and atomic, in contrast to the distributed lock. We proposed a standard of device lock for exclusive access of storage in the cluster environment. This command is to be included in the SCSI standard command set.

The STS system provides the sharing of tape devices in the SAN environment. Tape devices are serial, and thus the access to them must be exclusive. The backup task of one server must be finished before the backup tasks of other servers can begin. That is the reason for the exclusive lock implemented in the STS system.

Byte.Bit	7	6	5	4	3	2	1	0
0	Operation Code (83h)							
1	Reserved				Action			
2	(MSB)							
3	Lock Number							
4								
5	(LSB)							
6	(MSB)							
7	Client ID							
8								
9	(LSB)							
10	(MSB)							
11	Allocation Length							
12								
13	(LSB)							
14	Input Version Number (LSB)							
15	Control							

Fig. 4. Dlock Data

Figure 4 shows the format of the SCSI lock command. The operation code of the command is 83h. Figure 5 shows the meaning of the Action field. The STS only guarantees exclusive access among servers, so a device-based exclusive lock is needed instead of a data-block-based exclusive lock. The allocation Length field is not used.

A detailed description about the return format of the command can be found in[3]. A timer was added in the STS implementation to handle timeout. But in the case of backup, the backup time window may be quite long, so the timeout mechanism of the STS system needs to be refined. Considering the fact that the

Code	Action	Description	Data Returned
0h	Nop	No change, return current lock state	Type 1
1h	Lock Shared	Acquire shared lock	Type 1
2h	Lock Exclusive	Acquire exclusive lock	Type 1
3h	Force Lock Exclusive	Acquire exclusive lock, preempting lock if needed	Type 1
4h	Refresh Lock	Reset timer on lock	Type 1
5h	Unlock	Release lock	Type 1
6h	Unlock Increment	Release lock and increment version number	Type 1
7h	Activity On	Turn on Activity Monitor	Type 1
8h	Activity Off	Turn off Activity Monitor	Type 1
9h	Report Expired	Report which locks have expired	Type 2
Ah-Fh	Reserved	Reserved	

Fig. 5. Dlock Actions

backup time window in common backup systems is long and unpredictable, the policy of starting the timer on getting the lock was not adopted in the STS system. Every tape device maintains a log. Once a SCSI command from the server holding the lock is received, the timer is reset. So if a server is using the device for a backup task, it will hold the lock regardless of timeout. Termination of control occurs in two cases. When the backup task is finished, servers relinquish their resources. This is a normal way of state changing status. The other case is timeout. If the STS system has not received any command or message from the servers in a given interval, an unrecoverable error might have occurred. The STS system forces the server to relinquish the lock so as to free the resource.

The exclusive lock mechanism based on the STS system has the advantages of being platform-independent and fine-grained. It frees servers from communicating with each other, and therefore is more suitable for network storage systems.

4 Summary and Conclusion

This paper reported the implementation of a multi-server backup system based on the design described above. Redhat Linux and Windows 2000 were used in front-end servers. Ethernet and Fabricare were used for communication networks. The prototype of the STS was implemented in an I/O node machine. Backup software were Taper (Linux) [5] and TH-EasyBackup System (Windows) [6]. Backup tests showed that the multi-server backup system can be adapted to FC-SAN and IP-SAN systems and can support servers running different operation systems. The system adopts an exclusive lock mechanism based on device, which is a more reasonable mechanism for multi-server backup in the SAN environment, so the backup resources are used more effectively.

The STS system has the following features:

- (1) It shares the backup resources (tape devices) through a network. Servers running different OSs can access the tape device simultaneously.
- (2) A device-level exclusive lock mechanism can achieve the sharing of backup resources effectively and guarantee the correctness of backup tasks.
- (3) It has good compatibility, and supports various types of OSs, backup software, and networks (fibre channel and ethernet).

In short, the STS system is an effective multi-server backup system in the SAN environment. Moreover, considering the computing power of I/O nodes, the next approach is an intelligent interface between server and tape device, with which the STS system could take charge of backup tasks, and thus free the server resources for other use.

References

1. B.Phillips: Have storage area networks come of age?, IEEE Computer, vol.31, no.7, 10-12, July 1998.
2. Peter J. Braam. A Distributed Lock Manager for Linux. www.lustre.org/docs/dlm.pdf, 2003.
3. Kenneth W. Preslan, et al., Device Locks: Mutual Exclusion for Storage Area Networks, In the 7th NASA Goddard Conference on Mass Storage Systems and Technologies, March 1999.
4. ANSI, SCSI Stream Commands-2 (SSC-2), <http://www.t10.org>, Sep. 2002.
5. <http://sourceforge.net/projects/taper/>, 2004.
6. <http://storage.cs.tsinghua.edu.cn/>, 2004.
7. Douglas Gilbert, The Linux 2.4 SCSI subsystem HOWTO, <http://www.fsf.org/copyleft/fdl/html>, 2004.

Storage Virtualization System with Load Balancing for SAN

Weitao Sun, Ji-wu Shu, and Weimin Zheng

Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China
sunwt@tsinghua.edu.cn

Abstract. Logical Volume Manager (LVM) has been a key subsystem for online disk storage management. Additional layer is created in the kernel to present a logical view of physical storage devices. Many transparent functions can be implemented between the logical and physical layers, such as merging several physical disks into a larger logical device, resizing logical devices without stopping the system. In a logical volume group, files can be striped into several physical disks so as to achieve high I/O performance. But data I/O parallelism by itself does not guarantee the optimal performance of an application since higher data throughput does not necessarily result in better application performance. This paper studied the dynamic load balancing and data redistribution algorithms in the storage virtualization layer when the load becomes imbalanced across the disks due to access pattern fluctuation. An extension of the heuristic load balancing method was proposed to the storage virtualization subsystem of Tsinghua-Mass Storage Network System (TH-MSNS). Logical volume I/O request status is monitored and the physical disks are sorted according to the access number of Logical Extents (LE) per time unit. The I/O operations on a LE of the hottest disk are transparently migrated to other disks. The preliminary performance simulations under a WWW server file access workload give satisfactory results by the promising cooling algorithm in storage virtualization systems.

1 Introduction

Over the last decade, there has been a sustained explosive growth of Internet and data, which leads to a rapidly increasing demand for storage. Much effort has been elaborated on improvement of distributed storage to provide better performance and scalability. Storage Area Network (SAN) introduces a new scheme to reduce the workload of a file server by transferring data directly between the clients and the network storage system [1]. Disk arrays are used in SAN to provide mass storage and high performance I/O. Individual disks are combined into one logical volume by Logical Volume Manager (LVM) and can be used just like a real device. A request for a logical device and block must be mapped to a physical device and block for the low level driver by LVM. LVM will store data on the underlying devices in linear mapping or data striping. Striping data across

multiple disks has originally been proposed in [2] and [3]. Partitioning huge data object into small chunks and distributing them onto storage servers, adds some kind of parallelism that helps the client machine to achieve a better performance in handling the intensive data I/O. But parallelism by itself does not guarantee the optimal performance of an application since higher data throughput does not necessarily result in better application performance. The disk access rate and response imbalance fluctuate with time because of the different user access patterns. The works in [4], [5] and [6] presented file systems for dynamic data creation and reorganization in disk arrays. A dynamic file reallocation strategy was developed in [7] that adapts to a sequence of read and write requests whose location and frequencies are unpredictable. Another evolutionary algorithm for data allocation for distributed database systems was designed in [8]. However the study on dynamic load balancing of storage virtualization system is very limited. Since the storage virtualization is a very flexible layer and it works independent of any particular storage system, developing the load balancing data redistribution algorithms in LVM is being considered as a promising approach for SAN storage virtualization.

The work presented here aims at developing dynamic load balancing and data redistribution algorithms for storage virtualization system. Skewed I/O workload in the virtualized storage system can be balanced among all the storage devices in the volume group, no matter where the devices are or what type the devices are. The storage virtualization system load balancing promises to give a higher level performance improvement than any other disk array load balancing method. The heuristic load balancing method in [6] are adopted and extended to a storage virtualization version for TH-MSNS [9]. A WWW server workload generator and a simulated I/O system are implemented by CSIM [10] to provide more performance insights of this new algorithm.

2 Storage Virtualization for TH-MSNS

The TH-MSNS is an implementation of an FC-SAN. In the TH-MSNS, the storage node is composed of a general-purpose server, SCSI disk arrays, and fiber channel adapters. By using the SCSI target simulator [9], the storage devices attached by the storage node can be mapped to the host as its own local disks, on which the host's OS can directly create file systems and databases.

The storage virtualization subsystem of TH-MSNS collects the network attached storage devices into a large storage pool. This logical storage pool can be assigned to any clients with propriety size. The distributed storage virtualization system provides a consistent layout to all the servers so as to keep the coherence of their kernel data. The system framework is composed of file system, the distributed virtualization system kernel, the distributed virtualization system management module, configure module, synchronization module, communication module and the center control module in the management node (Fig. 1). The distributed storage system provides many online volume management features, such as logical volume add/delete, logical volume resizing, snapshot and online backup.

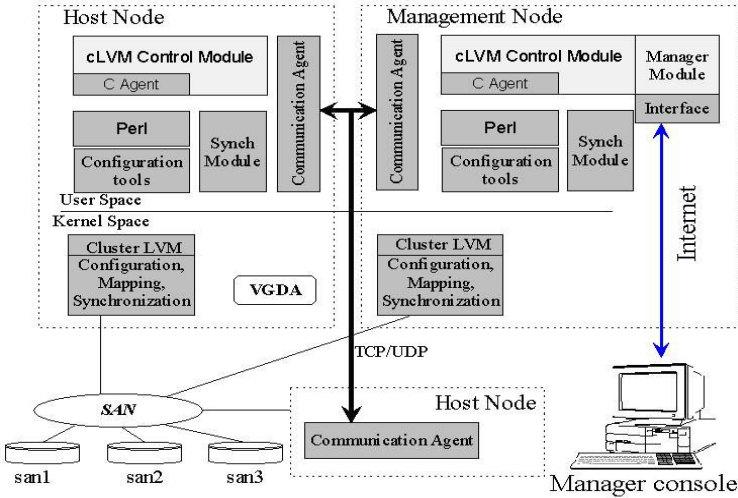


Fig. 1. The distributed storage virtualization system of TH-MSNS

3 Load Balancing for TH-MSNS Storage Virtualization

The load balance of a distributed storage system depends on the data distribution, regardless of whether the files are partitioned or not. The data in virtualized storage system are redistributed according to the access pattern fluctuation. The lower level in the linux LVM storage hierarchy is the *Physical Volume* (PV). Each PV is divided into equally sized *Physical Extents* (PE). The size of PE is variable but equal for all PEs in a VG. A PV is a single device or partition with a Volume Group Descriptor Area (VGDA) on it. The volume manager put the PVs into storage pools called *Volume Groups* (VG). A VG is the equivalent of a physical disk from the system viewpoint. VG is the storage pool from which *Logical Volumes* (LV) can be allocated. LVs are the actual block devices on which file system can be created. Every LV is divided into *Logical Extents* (LE). LEs are of the same size as the PEs of the VG the LV is in. Every LE is mapped to exactly on PE on the PV.

3.1 Heat Tracking of Virtualized Storage System

In order to perform the online data reorganization without system suspending, it is necessary to estimate the request size and frequency to different LEs and LVs. The *heat* and *temperature* are used as statistical parameters. The *heat* of LEs and LVs is defied as the sum of access number of a LE or LV per time unit. It is determined by statistical observation over a certain period of time. The *temperature* of a extents is defined as the ratio between heat and size.

The heat is tracked by the reciprocal of last k requests average interval time. Above heat tracking method is very responsive to sudden increase in an extent's

heat. However, the method has difficulty to deal with a sudden heat drop. An “aging” method [6] was introduced for heat estimates. Simulated “pseudo requests” are periodically invoked to all logical extents. Whenever such a pseudo request would lead to a heat reduction, the heat estimate is updated.

The heat lists of the physical disks that constitute the virtualized storage system are computed through the mapping information. When an application wants to access storage on a logical volume, the LE is identified. By using the unique ID number of the LE in the LV, both the PV and the PE are found in the mapping table. Then the access frequency of this LE can be associated with the PE and the heat of the physical disk can be calculated. Here the physical disks are fully used by the virtualized storage system.

3.2 Load Balancing of Virtualized Storage System

An optimized disk cooling must provide a good compromise between maximizing I/O performance of the virtual storage system and minimizing the work invested in data reorganization. Extents are removed from the hottest physical disk to obtain the maximal gain while the additional I/O costs are kept in a low level. The temperature based extents selection algorithm in [6] is extended to the virtualized storage system. The pseudo program of the basic cooling algorithm for TH-MSNS virtual storage system is illustrated as follows.

```

Input:   D    -number of PVs in the virtualized storage system.
          HLj  -heat of LE j                      HPi  -heat of the PV i
          H'   -average PV heat
          Ei   -list of LE on PV i in descending temperature order
          Ei'  -list of LE on PV i in ascending temperature order
          D'   -list of PVs in ascending heat order

Step 0: Initialization: destination = notfound
Step 1: Select the hottest PV s
Step 2: if HPs>H'(1+delta) then
Step 3: while (Es not exhausted)and(destination==notfound) do
          Select next LE e in Es
Step 4: while (D' not exhausted)and(destination==notfound) do
          Select next PV t in D' in ascending heat order
          if (t not hold LE of the file to which e belongs)then
              while (Et' not exhausted)and(destination==notfound)do
                  Select next LE e' in Et'
                  if(HLe>HLe') destination=found fi
              endwhile
          fi
          endwhile
        endwhile
Step 5: if s has no queue then
          HPi* =HPi-HLe+HLe'    HPt* =HPt+HLe-HLe'

```

```

    if HPt* < HLe then
        reallocate LE e and e' between PV s to PV t
        HPs=HPs*          HPt=HPt*
    fi
fi
fi

```

4 Experiment with WWW Workload

To study the effects of load balancing in TH-MSNS storage virtualization subsystem for realistic application settings, extensive experiments are conducted based on access traces that contain one month's worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The log was collected from 00:00:00 July 1, 1995 to 23:59:59 July 31, 1995, a total of 31 days. Almost 22450 files are requested nearly for 1891713 times with heavily skewed access frequencies. The access frequency variations at different load level are carried out by a “speeding up” method. Original requests interval time is multiplied by a speeding up factor $1/\lambda$.

In this experiments 10000 files of two types of sizes are used. The files sizes are hyperexponentially distributed such that each file belongs to different class with a mean size of either 20 MBytes or 40 MBytes. Three PVs each with a capacity of 20 GBytes are virtualized as a VG composed of five 12GBytes logical volumes in a striped manner (Fig. 2). The LE and PE sizes are 4 MBytes. All the 10000 files are randomly distributed in the two logical volumes. The expected service time T_{serv} for each request on the track of disk i is given by:

$$T_{serv} = t_{start}^i + t_{seek}^i \cdot \Delta + t_{trans}^i. \quad (1)$$

where t_{start}^i , t_{seek}^i , Δ and t_{trans}^i denote the arm start time, arm track seek time, the track difference between previous and current arm position, the transfer time, respectively. In this experiment, the three physical volumes have the same parameters introduced in Eqn. 1. The start time t_{start}^i is 0.005 second. The track seek time t_{seek}^i is 0.0001 second. The transfer time t_{trans}^i is 0.01second.

The cooling was invoked every 10 seconds and the imbalance threshold $\delta=0.2$. Here the file access frequency speeding up factor λ is considered as a workload parameter. Fig. 3(a) shows the response time curves fluctuated with time of the three physical volumes without cooling when the factor λ is 80. Fig. 3(b) shows the response times fluctuation with cooling. Fig. 4 (a)-(b) shows the PV utilizations without and with the proposed cooling method. Fig. 5 shows the total cooling steps at different times.

Though the volume group is formed in a striped manner and the files are randomly distributed in the logical volumes, the dynamically evolving HTTP file access patterns cause a skewed workload on the virtualized disks as shown in Fig. 3. This load imbalance is especially high at the early time, because only a small mount of files are visited during this short period. As time goes on, more files are access in the virtualized disks and the I/O workload imbalance become

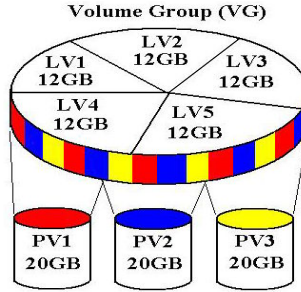


Fig. 2. The virtualized disk configuration composed of three Physical Volumes (PV) in striped manner

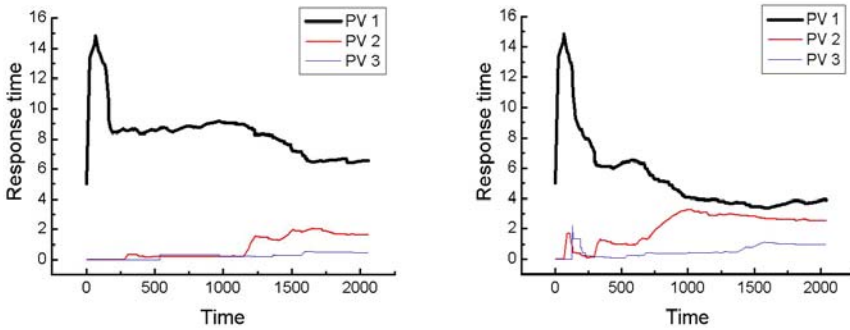


Fig. 3. The response times of three PVs at different time: (a) (left figure) without cooling method (b) (right figure) with cooling method

relatively steadier. Since the PV1 is much busier than PV2 and PV3, the PV1 response time curve in Fig. 3 is higher than the other two. PV3 response time is close to zero, which means this physical volume is under loaded and the hot logical extents should be reorganized between three PVs.

The response time figures indicate that access skew does have a disastrous effect on performance. The underlying reason is that the hottest disk has a much higher utilization than the overall virtualized disk system and forms a bottleneck. The cooling algorithm exhibits noticeable response time improvement by an order of 2 times. For $\lambda = 80$, the PV utilization curves are shown in Fig. 4. The average and minimum utilization of the hottest PV was 97.57% and 94.5% without cooling. With cooling, the average and minimum utilization of the hottest PV was reduced down to 82.08% and 66.9%. As a payment for response time improvement, the average utilization of the whole virtualization disks system increased up from 37.63% to 44.54%. Note that an average utilization of 44.54% appears to be a relatively light load.

Fig. 5 shows the cooling frequency variation over the duration of the experiment. The cooling is invoked periodically when the hot PVs and LEs need

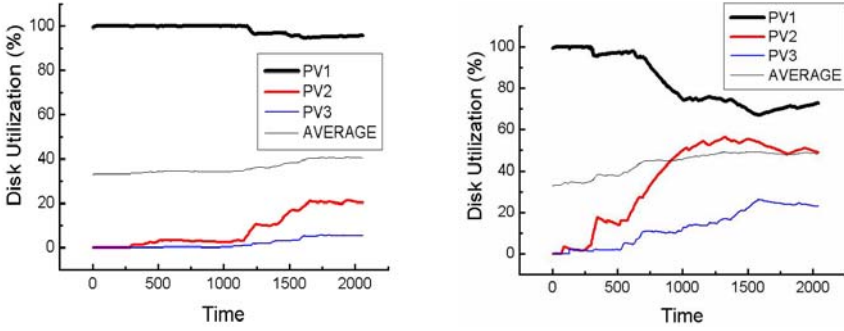


Fig. 4. The utilizations of three PVs at different time: (a) (left figure) without cooling method (b) (right figure) with cooling method

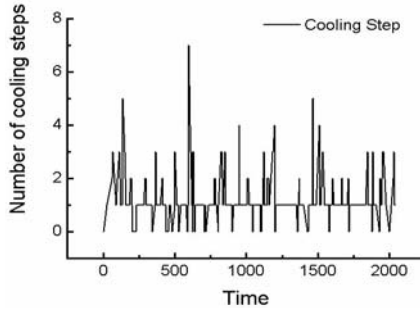


Fig. 5. The cooling step number at different time

dynamic reorganization. At these points the method without cooling suffers from long disk queues because of the load imbalance. The cooling method only need several cooling steps to balance the workload and is fairly successful to achieve response time improvements. The experiment results give a solid support to this storage virtualization load balancing method. System performances are improved under medium and heavy workload by an order of two times.

5 Conclusions

Based on storage virtualization subsystem of TH-MSNS, a workload-balancing algorithm is developed to improve the I/O performance of the virtualized disks. The heat of physical volume and logical extents are tracked and the hottest logical extents are reorganized in the virtualized storage system to balance the I/O request load. The cooling procedure is periodically invoked and will be triggered if the load imbalance exceeds the threshold. Simulated experiment with NASA WWW server HTTP requests I/O data shows that the cooling method indeed cut down the response time of the virtualization system. The load balancing in virtualized system is transparent to the clients and can be carried out among a wide variety of storage devices.

Acknowledgements

The work described in this paper was supported by the National High-Tech Research and Development Plan of China under Grant No. 2001AA111110.

References

1. Wilson, S.: Managing a Fibre Channel Storage Area Network. Storage Networking Industry Association, November (1998)
2. Kim, M. Y.: Synchronized Disk Interleaving. IEEE trans. On computers, Vol. C-35 No. 11 (1986)
3. Salem, K. and Garcia-Molina, H.: Disk Striping. Proc. Of the 2nd Int. Conf. On Data Engineering (1986)
4. Weikum, G., Zabback, P. and Scheuermann P.: Dynamic File Allocation in Disk Arrays. ACM SIGMOD Conf. (1991)
5. Tsai, H. P., Huang, J. L., Chao, C. M., et al.: SIFA: A Scalable File System with Intelligent File Allocation. Proc. of the 26th Int. COMPSAC (2002)
6. Scheuermann, P., Weikum, G., and Zabback, P.: Data Partitioning and Load Balancing in Parallel Disk Systems. VLDB J. 7(1)(1998) 48-66
7. Awerbuch, B., Bartal, Y. and Fiat, A.: Competitive Distributed File Allocation. Information and Computation, 185 (2003) 1-40
8. Ahmad, I., Karlapalem, K. and Kwok, Y. K. et al.: Evolutionary Algorithms for Allocating Data in Distributed Database Systems. Distributed and Parallel Databases, 11 (2002) 5-32
9. Shu, J. W., et al.: A Highly Efficient FC-SAN Based on Load Stream. In: Zhou, X.M., Jahnichen, S., Xu, M. and Cao, J.N. (eds.): The Fifth International Workshop on Advanced Parallel Processing Technologies, LECTURE NOTES IN COMPUTER SCIENCE 2834 (2003) 31-40
10. Schwetman, H.: CSIM19: A Powerful Tool for Building System Models. In Proceedings of the 2001 Winter Simulation Conference. Ed. B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Roher, Washington, D.C. (2001) 250-255

Design and Optimization of an iSCSI System

Bigang Li, Ji-wu Shu, and Weimin Zheng

Tsinghua University, Beijing, China
lbq01@mails.tsinghua.edu.cn

Abstract. Data storage plays a critical role in today's fast-growing data-intensive network services. iSCSI is a new standard that allows SCSI protocols to be carried out over IP networks. This paper introduces a software iSCSI implementation and proposes two mechanisms to improve the performance of the IP SAN. One is the appropriate algorithm to manage commands on the SCSI command queue, such as an elevator algorithm to prioritize commands, and algorithms to eliminate or concatenate SCSI commands. The other mechanism is the use of a cache algorithm on the iSCSI target server. We have implemented these optimized algorithms and tested the performance of the IP SAN. The results show that the optimized system's throughput reaches 85MB/s, which consumes 90% of the total IP network bandwidth and greatly reduces the average response time. Our IP storage system shows improved performance compared with the current iSCSI implementation.

1 Introduction

Storage Area Networks (SANs)[1] use a net-oriented storage structure, which enables the separation of data processing and data storage. SANs have the virtue of high availability and scalability, high I/O performance, and data sharing. Internet SCSI, or iSCSI is a new standard[2], which encapsulates a SCSI subsystem within a TCP/IP connection. The IP SAN can now be implemented using less expensive components, so it has become widely popular in recent years.

The IP SAN has high price/performance ratio and the IP network is ubiquitous. Furthermore the IP network is cheap to build and manage. The software iSCSI (or IP SAN) can transfer the SCSI protocol very well, but the IP SAN does not have very high performance for various reasons. A majority of such small packet traffic over the IP network lowers the network utilization.

Recently, many software iSCSI implementations have been developed, such as IBM Haifa Research Lab[3], University of New Hampshire[4] and Intel[5]. They all have the basic functions of the iSCSI system. Some of them have even released their source code for researchers[6]. But none of them focus on the optimization of the iSCSI system or tries to bridge the disparities between the SCSI and IP protocols. Xubin He[7] and his colleagues have attempted to improve current iSCSI performance by implemented an iSCSI cache system only on the initiating server. Their work did not include on the target server of the iSCSI system.

The idea of using cache technology to improve performance has been used in both file systems and database systems for many years, such as the Disk

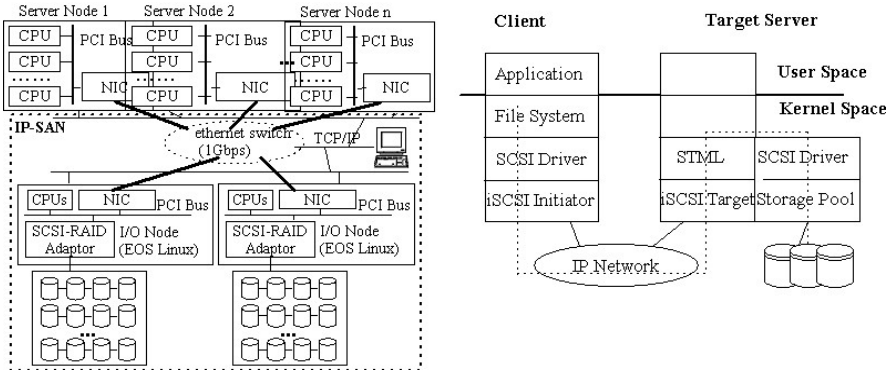


Fig. 1. The Architecture of our IP SAN

Caching Disk (DCD)[8].Several Redundant Array of Inexpensive/Independent Disks (RAID) systems have implemented the LFS algorithm at the RAID controller level[9][10],but implementing the cache system on the target server and applying the special optimization techniques for the SCSI command queue are novel approaches.

This paper introduces a software iSCSI implementation and proposes two innovations to improve the performance of the IP SAN. One is the appropriate algorithm to manage commands on the SCSI command queue, such as an elevator algorithm to prioritize commands, and algorithms to eliminate or concatenate SCSI commands. With these optimization designs, the storage system can input or output more data during a given time. The other innovation is to utilize a cache algorithm on the iSCSI target server. The cache system can bridge the disparities between the SCSI and IP protocols. It converts small commands or requests into large ones before writing data onto physical disks, and utilizes the log structure to quickly write data to log memory for caching data. Moreover, the cache system is completely transparent to the operating system, so no changes to the operating system are required.

We have implemented these optimized algorithms in Linux based on the TsingHua Mass Storage Network System (TH-MSNS)[11]and tested the performance of the IP SAN. We use the Iometer[12] software to test the performance. The results show that our IP storage system demonstrated improved performance compared with the current iSCSI implementation.

2 Architecture

Figure 1 shows the whole hardware architecture of our IP SAN. The server node acts as an initiator in the iSCSI protocol, and the I/O storage node is the target server. The initiators send the SCSI commands/messages to the target server through an IP network. The iSCSI target module and the SCSI Target Middle Level (STML) module on the target server receive them. After the com-

mands/messages are executed, the STML sends an echo or data to the initiator server through the IP network. In the IP SAN system, the initiator share storage resources at the device level. Moreover, the servers share a storage pool at the file system level in Network Attached Storage (NAS)[13][14].

There are three main modules in the iSCSI system. They are the iSCSI initiator module, the iSCSI target module and the SMTL. They all work in the kernel space of the operating system. The iSCSI initiator module registers some remote network disks to the initiator node's file system and sends the SCSI commands/messages to the target. The target module receives the SCSI commands/messages from the IP network and transfers them to the STML. Our optimization research also focused on the STML. The detail progress is showed in ref[4].

3 Optimization of the iSCSI System

iSCSI systems don't perform very well for various reasons. In this section we propose two ways to improve the performance of the storage system.

3.1 Optimization Technologies on the SCSI Command Queue

Often the STML module keeps thousands of SCSI commands at a time. If the STML treated these commands in order, the average response time would be very long. On the other hand, the iSCSI target server is often idle because of low CPU utilization. The server may use the optimization technologies to speed up the I/O performance such as a journal log file system, but the iSCSI target server receives SCSI commands from many initiator servers and arranges these command into the queue, so there are many chance to optimize the SCSI command queue to speed up the iSCSI system performances. Before we explain these algorithms, some symbols will be introduced.

S_i is assumed for the i -th SCSI command. The SCSI command has several attributes, as described below.

A_i : the operation address (sector) of the SCSI command.

O_i : the attribution of the SCSI command. Its value might be R (read data), W (write data) or N (no data to transfer).

The Elevator Algorithm for the SCSI Command Queue. If the STML executed the SCSI command in order, the magnetic needle of the physical disks would move from place to place on the disk, taking a long time to execute commands. In one I/O operation, the main reason for the slow response latency is the time required for the magnetic needle to move. The commands in the queue are immethodical because they come from different initiator servers. If we arrange the order of the SCSI commands, the performance will improve[15]. The algorithm that organizes the commands is known as the elevator algorithm. Moreover, the elevator algorithm for the SCSI command queue is not right every

time. Any change for the queue could be carried out through the exchange of two elements, so it is only necessary to consider the exchange of two SCSI commands.

When two SCSI commands fit the condition below, they can exchange positions correctly.

$$(A_i \cap A_j == \phi) \parallel ((O_i == R) \& \& (O_j == R))$$

Otherwise the executing order for commands i and j can not exchanged. Our algorithm follows this rule to regulate the order of the SCSI command queue for better performance.

Concatenation of the SCSI Commands. There are two or more SCSI commands operating the continued sectors due to the Locality Principle of the application. The STML module could concatenate these two SCSI commands into one to save executing time. The condition required for two commands to be concatenated into one is expressed as:

$$(\max(A_i) == \min(A_j)) \& \& (O_i == O_j)$$

In other words, when the two SCSI commands have the same attribute and the operating address is continuous, they can be concatenated. Furthermore, when A_i and A_j cross each other, they can be concatenated also, even though this case seldom occurs.

Elimination of the SCSI Commands. If we analyze the SCSI command queue carefully, we find that some SCSI commands can be eliminated. The condition for one SCSI command to be deleted is somewhat complicated. The SCSI commands between i and j should all be considered. The condition is explained below.

Let $j > i$, if

$$(O_i == O_j == W) \& \& (A_i \subseteq A_j)$$

is true, the STML can eliminate command i . Furthermore,

(1) if $\forall k, i < k < j, A_k \cap A_i = \phi$, the STML can eliminate the command directly, and

(2) if $\exists k, i < k < j, (A_k \cap A_i \neq \phi) \& \& (O_k == W)$, the STML needs to keep the data of command i until command k has been executed correctly.

After the eliminate the command i , system would just return one success (DID_OK) response to the initiator server.

The STML can use the three algorithms for the SCSI command queue at the same time, but the elimination of the SCSI commands and concatenation of the SCSI commands should be considered firstly.

3.2 The Cache System

We used the memory on the target server as the cache for the iSCSI system. figure 2 shows the structure of the cache, which contains three main parts: Head Information, Bitmap Table and Block Data.

The Head Information is made up of many elements, and every element has its own unique ID. Each element is for a SCSI command to cache data. As shown in figure 2, *CacheAddr* means the address in the cache system, and it is an integer. It always points to the Block Data. The *DataLength* indicates the length of the cache data. The *CacheAddr* and the *DataLength* determine an area for the SCSI command to store data. The field Time indicates the time of the SCSI command, while the *LBA* means *Logical Block Address*. Status indicates the status of the current data block. It might be *Valid*, *Dirty* or *None*. If the value of the Status is *Valid*, it means that the current block data in the memory is accordant with the data on the disks. Dirty has the opposite meaning, and *None* indicates that the current memory has no valid data, or is not being used by any SCSI command. So if the status of an element is *None*, this element can be used by a subsequent SCSI command.

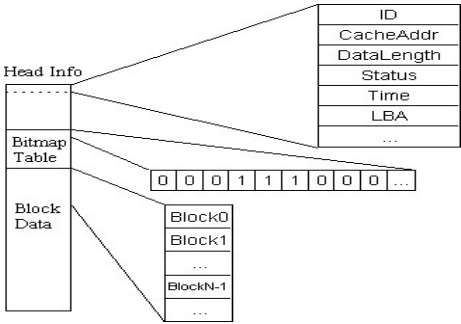


Fig. 2. Organization of the cache system

Another question is the cache system provides persistency. If the environment required the high availability, the RAM should use battery-backed RAM to save the cache data when the power down suddenly.

I/O Flow of the Cache System. When the cache system receives a new write operation, it searches the corresponding element through the hash table. If another SCSI command is received before the corresponding element is found, the data for the new command is sent to the data block indicated by the element, and the old data is erased. At the same time, the cache system changes the status of the element to *Dirty*. If the element is free, the cache system copies the write data to the cache system, and sends the *ACK* information for the write operation. When the whole cache system is full, the synchronization operation is executed immediately. After the synchronization operation, all the elements in the *Head Information* and the block data are free. The flow of the read operation is simpler than the write operation's.

Synchronization of the Cache System. The synchronization of the cache system is very important for data protection. In our cache system, one kernel

thread is designated to synchronize the data technically. This thread is idle most of the time, but it is awaked if one of the following events occurs:

- (1) The Head Information of the cache system is full, and there is no element for the new SCSI command.
- (2) The Block Data is full.
- (3) The cache system is idle. The cache system has not received any new SCSI commands for a long time.
- (4) The cache system receives the synchronized command sent by the manager.
- (5) The cache system will be uninstalled before long.

During the synchronization operation, the block data is out of service; otherwise the new command would be likely to write dirty data to the Block Data. After the synchronization operation, the Block Data is free for new SCSI commands.

4 Performance Evaluation

In order to evaluate the new system's performance, we implemented the iSCSI system and its optimization system prototype based on the TH-MSNS. We compared the iSCSI system's performance with and without the optimization system. To ensure an impartial comparison, we tested the iSCSI system using the same hardware configuration. The bandwidth of the IP network for these computers was 1Gbps, and the cache memory size for the cache system was set as 128MB. The CPU was Intel Xeon 2.4G and disks are 10k rpm Seagate SCSI disks (73GB). We used two nodes as the clients and one as the target server. The ethernet cards are all Intel e1000. We measured the system throughput using Iometer[12].

Figure 3 shows the throughputs of the read and write operations on the iSCSI system. The results demonstrate that the optimization system greatly enhances the performance of the iSCSI system. The white column shows the iSCSI performance with the optimization technologies on the SCSI command queue. The blue column shows the iSCSI performance with both the optimization technologies on the command queue and the cache system. With the optimization system, the average data throughput reaches 85MB/s. Furthermore, with increases in the transfer size, the gap between the two iSCSI systems' performance decreases. This could be because of the limitations of the IP network. Another factor is that when the transfer size is small, the SCSI command queue is long and the optimization is more effective. If the command queue is very long, the chance of deleting or linking some SCSI commands is very frequent. Moreover, the elevator algorithm would be also more effective. If the SCSI command queue is not very long, the cache system is almost completely responsible for the optimization of the iSCSI system. When the throughput of the system reaches 80MB/s, there is not much space for the performance to speed up, since the IP network can only carry data at that speed.

From the Figure 3, the iSCSI optimization system causes the CPU utility to rise from around 6% to 7%. This means that the optimization system only uses



Fig. 3. This figure shows the test results of IP-SAN. The read/write throughput, CPU utilization and the average response time were measured

the CPU a little, and the CPU is still mostly idle. Moreover, the results show that the optimization system only uses a small amount of CPU time, while the effect is very well contrastively.

The data in figure 3 shows that without the optimization system, all of the data must be transferred from the physical disks, so the average response time is very high. It shows that more than 90% of the I/O operation latency is more than 4000 microseconds. But with the optimization system, more than 60% of the I/O operation latency is less than 4000 microseconds. This shows that the optimization system very effectively educes the latency of the I/O operation system.

5 Conclusion

In this paper, an iSCSI system and its optimization system is introduced in detail. We explained some optimization technologies and proposed two strategies to improve the performance of the IP SAN. One is using the proper algorithm on the SCSI command queue, and the other is using a cache algorithm on the iSCSI target server. The algorithm were tested, and results using the Iometer benchmark showed shows that the improved iSCSI system has high performance, and the throughput reaches 90% of the IP network's bandwidth. The latency of I/O operations is reduced greatly. The results demonstrated that the proposed optimization technologies greatly improve the iSCSI system's performance.

Acknowledgement

The work described in this paper was supported by the National High-Tech Research and Development Plan of China under Grant No. 2001AA111110. We are grateful to our colleague Pan Jiaming and Mu Fei for their encouragement.

References

1. B.Phillips: Have storage area networks come of age?, IEEE Computer, vol.31, no.7, 10-12, July 1998.
2. Julian Satran, et al.: iSCSI standard, <http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.pdf>, 2003.
3. Kalman Z. Meth: iSCSI Initiator Design and Implementation Experience, 10th Goddard Conference on Mass Storage Systems and Technologies, 2002 (see <http://romulus.gsfc.nasa.gov/msst/conf2002/toc.html>).
4. Ashish Palekar: Design and Implementation of A Linux SCSI Target for Storage Area Networks, Proceedings of the 5th Annual Linux Showcase & Conference, 2001.
5. Intel Corp: Intel iSCSI project, <http://sourceforge.net/projects/intel-iscsi>, 2001.
6. <http://sourceforge.net/projects/unh-iscsi/>, 2004.
7. Xuebin He, Qing Yang, Ming Zhang: A Caching Strategy to Improve iSCSI Performance, Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, 2002.
8. Y. Hu and Q. Yang: DCD-disk caching disk: A New Approach for Boosting I/O Performance, Proc of 23rd Annual Intl. Symposium on Computer Architecture (ISCA'96), 1996.
9. Y. Hu, Q. Yang, and T. Nightingale: RAPID-Cache –A Reliable and Inexpensive Write Cache for Disk I/O Systems, Proc. of the 5th International Symposium on High Performance Computer Architecture (HPCA-5), Orlando, Florida,1999.
10. J. Menon: A Performance Comparison of RAID-5 and Log-Structured Arrays, Proc. Of 4th IEEE Int'l Symp. High Performance Distributed Computing, 1995.
11. Tsinghua-Mass Storage Network System report, The department of computer science and technology, Tsinghua University, 2002.
12. Intel Corporation, <http://www.iometer.org/>, 2003.
13. G. Gibson and R. Meter: Network Attached Storage Architecture, Communications of the ACM, vol. 43, 2000.
14. R. Hernandez, C. Kion, and G. Cole: IP Storage Networking: IBM NAS and iSCSI Solutions, Redbooks Publications (IBM), 2001.
15. L. Huang and T. Chiueh: Implementation of a rotation-latency-sensitive disk scheduler. SUNY at Stony Brook Technical Report, May 2000.

Measurement and Modeling of Large-Scale Peer-to-Peer Storage System

Gang Liu, Mingzeng Hu, Binxing Fang, and Hongli Zhang

Department of Computer Science and Engineering, Harbin Institute of Technology
Heilongjiang 150001, China
{Lg, Mzhu, Bxfang, Zh1}@pact518.hit.edu.cn

Abstract. The peer-to-peer applications have become the killer applications in information share internet revolution. It is therefore important to analyze and evaluate network topology and its corresponding geometric characteristic of the overlay network. The peer-to-peer systems based on restricted flooding mechanism, typically Gnutella, are still the most popular peer-to-peer systems. Recently a few main peer-to-peer applications, such as BearShare, Limewire and so on, which constructed the new Gnutella network 0.6, are implemented based on Gnutella Protocol version 0.6 that makes much improvement upon version 0.4, while early measurements of peer-to-peer systems largely aimed at the systems built based on version 0.4. The paper develops a new “network crawler” to extract the topology of Gnutella network 0.6, analyze the topology graph, evaluate corresponding static geometric characteristic, and build its network mechanism model, including small-world and power-laws model.

1 Introduction

The emergence of novel network applications such as Napster [1], Freenet [2], and Gnutella 0.4 [3] has reincarnated the familiar peer-to-peer (P2P) architecture model of the original Internet in new and innovative ways in an effort to facilitate world-wide sharing of information. These applications are mainly designed and used for large-scale sharing of audio and video files. In such systems, end-hosts self-organize into an overlay network and share content with each other. Compared to the traditional client-server model, files are served in a distributed manner and replicated among the network on demand. Since hosts participating P2P networks also devote some computing resources, such systems scale with the number of hosts in terms of hardware, bandwidth, and disk space. The world-wide popularity of succedent iMesh [4], eDonkey [5], Bittorrent [6], KaZaA [7] and Morpheus [8] based on FastTrack, Bearshare [9] and Limewire [10] based on Gnutella 0.6, implies that the P2P applications have become the killer applications in information share internet revolution. The stunning growth and the bandwidth intensive nature of such applications suggests that P2P traffic can have significant impact on the underlying network. It is therefore important to analyze and evaluate network topology and its corresponding geometric characteristic of the overlay network, understand and characterize this traffic [12] in terms of end-system behavior and network impact in order to develop workload models and provide insights into network traffic engineering and capacity planning.

The success of this revolution will depend on the ability of modern P2P network application to provide efficient communication between increasingly large number of autonomous hosts dispersed all over the Internet. To cope with this problem some P2P applications, like instant messaging and Napster rely on a centralized server, while the applications, such as eDonkey, FastTrack and Bittorrent, use multiple centralized index servers to construct cluster-based P2P architecture [14]. Other applications, such as Gnutella and Freenet, adopt fully decentralized design approach, require scalable algorithmic solutions for functions such as routing and searching. Fully decentralized P2P storage system builds a pure P2P architecture and can factually reflect the characteristics of P2P network topology. To consider the less number of nodes joining Freenet and Gnutella 0.4 [13], this paper choose Gnutella 0.6 as the measured P2P storage system.

2 Methodology

The methodology behind our measurements is basically the design of the Gnutella Crawler which discovers Gnutella network topology. Compared with IP networks, Gnutella network is highly dynamic. This means that its topology is constantly changing - nodes and edges are added and removed as hosts join and leave the network, establish new connections, and close the existing ones. Therefore any topology discovery algorithm operating on the Gnutella network is really capturing an instance, or a snapshot of the topology at a specific point in time. Clearly, this produces an additional requirement for any topology discovery algorithm to be efficient, since the accuracy of the topology map is inversely proportional to the actual running time of an algorithm that was used to obtain it. In designing the crawler, the paper has paid close attention to this requirement.

2.1 Gnutella Architecture and Protocols Evolution

Gnutella is an overlay network superimposed on top of the Internet. Gnutella network architecture consists of a dynamically changing set of nodes connected using TCP/IP protocol. Every node (Servent or Peer) acts as a client who originates queries, and a server that provides file information and acts as a router. A Gnutella network consists of a set of interconnected nodes, at any given point in time. A new Gnutella user starts an instance of the Gnutella node software and the node uses out-of-band means to locate another node and establish a connection to it. This extends the net and makes the new node's files available to all other nodes in the net. Once connections are established, nodes use the Gnutella protocols to communicate. There is an initialization conversation following which nodes send out typed packets into the Gnutella network to locate and retrieve files. Gnutella network was initially established by using protocol version 0.4, and now protocol version 0.6 have replaced old version 0.4 as Gnutella network essential protocol. But there are still a small part of nodes performing version 0.4 protocol on Gnutella network.

Gnutella 0.6 makes much improvement upon self-organization and flow-control of Gnutella network, and further more offers some important protocol extensions.

Standard Message Architecture adopted by Gnutella 0.6 adds a new "Bye" Message and is compatible with Gnutella 0.4 in other Message format. In the self-organization aspects, Gnutella 0.6 first introduces the new web caching system, predominant Bootstrapping. The goal of the "Gnutella Web Caching System" (the "cache") is to eliminate the "Initial Connection Point Problem" of a fully decentralized network. Originally, all Gnutella nodes were connected to each other randomly. It worked fine for users with broadband connections, but not for users with slow modems. The problem can be solved by organizing the network in a more structured form. Gnutella 0.6 adopts an Ultrapeer system which has been found effective for this purpose. It is a scheme to have a hierarchical Gnutella network by categorizing the nodes on the network as leaves and ultrapeers. Gnutella 0.6 uses an extensible handshaking protocol for the Gnutella network. The handshaking scheme uses HTTP-style headers including two crucial headers X-Try and X-Try-Ultrapeers for Gnutella Crawler developments.

2.2 Data Collection: The Gnutella Crawler Design

Topology discovery in IP networks [17] is a well-studied area of research. Generally the approach is based on some protocol-specific feature, as in the case of traceroute. Although Gnutella protocol is much simpler than IP and provides no feedback regarding message delivery, it nevertheless provides the necessary functionality for mapping Gnutella network topology [15]. Notice that, according to the Gnutella protocol, it is possible to discover neighbors of a particular host.

The paper has developed a crawler (see fig 1) that joins the Gnutella network as a servent and uses a mechanism for combining active probing with passive listening to collect topology information.

1. The crawler starts with a list of web cache URLs which are got from the known Gnutella Web Cache websites and the up-to-date popular Gnutella applications such as Bearshare, Limewire, Gnutella-gtk and Mutella. Through accessing some URL, the active prober can obtain two kinds of data: other URLs registered in the website directed by the URL and some bootstrapping hosts. The prober adds these URLs into web cache list, and these bootstrapping hosts into host cache list.
2. The prober connects to all hosts in host cache list, uses PING/PONG interactive mechanism and Message headers X-Try and X-Try-Ultrapeers information in Handshaking process to acquire neighbor information of each host, and save adjacency host pairs in adjacency table.
3. The sniffer passively listens the Gnutella network, receives all connect request, and save the corresponding request hosts into host cache list.

For obtaining much more hosts, the upper three steps are performed repeatedly. To consider the demand of getting a snapshot of the Gnutella network as possible as quickly, the crawler implements multithread and asynchronous I/O operation.

3 Measurement Results Analysis

After measurement each time, the Gnutella crawler gets a topology graph of the Gnutella network, including all hosts (nodes) joining the network, and their adjacency

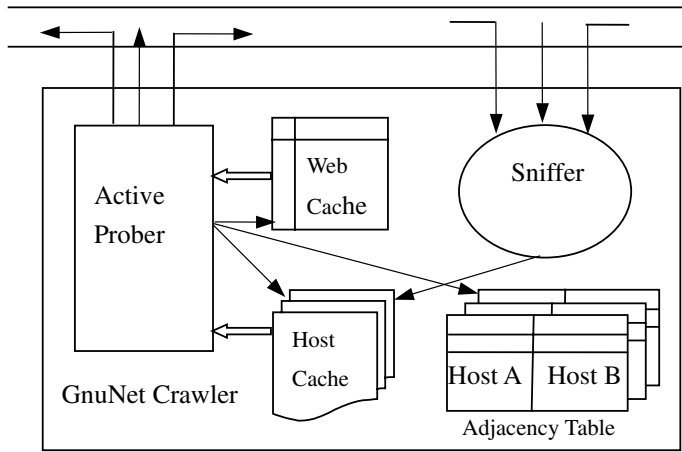


Fig. 1. The Gnutella Crawler Function

relationship. The topology graph is denoted an undirected graph. The metrics that have been used so far to describe graphs are mainly the node outdegree, and the distances between nodes. Given a graph, the outdegree of a node is defined as the number of edges incident to the node. The distance between two nodes is the number of edges of the shortest path between the two nodes. Most studies report minimum, maximum, and average values and plot the outdegree and distance distribution.

3.1 Small-World Modeling

The small-world phenomenon in the context of a worldwide social network refers to a widely accepted belief that we are all connected by a short chain of intermediate acquaintances. Some known networks have shown their small-world characteristics. Watts and Strogatz [16] define small-world behavior in terms of two properties, mainly the characteristic path length and clustering. In order to quantify these properties for various networks, the two defined characteristic path length L and clustering coefficient C as the following:

Definition 1. Characteristic path length L , a global property, is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices.

Definition 2. Clustering Coefficient C_v , a local (node) property measuring “cliquishness” of vertex v , is calculated by taking all the neighbors of v , counting the edges between them, and then dividing by the maximum number of edges that could possibly be drawn between those neighbors. Clustering coefficient C of a graph is defined as the average of C_v over all vertices v .

The results clearly demonstrate the small-world phenomenon for these networks: $L \geq L_{random}$ but $C \gg C_{random}$ [16], L_{random} and C_{random} denote characteristic

path length and clustering coefficient, respectively. Upon analyzing the Gnutella network topology data obtained by our crawler, we discovered both the small diameter and the clustering properties characteristic of small-world networks. To show this, we calculated the clustering coefficient and the characteristic path length as defined by Watts and Strogatz for five different snapshots of the Gnutella topology obtained during the months of December 2003 and April of 2004 (see Table 1).

Table 1. Statistics for five snapshots of the Gnutella network topology

Snapshot date	Nodes	Edges	Diameter
2003.11.20	98127	861779	9
2004.01.08	102489	772036	9
2004.01.25	86305	569808	9
2004.02.16	111258	939573	9
2004.03.25	122089	1199901	8

As you can see, all of the Gnutella topology instances show the small-world phenomenon: characteristic path length is comparable to that of a random graph (see table 3), while the clustering coefficient is considerably higher (see table 2). These results clearly indicate strong small-world properties of the Gnutella network topology. $G(N,p)$ denote the random graph with N vertices, every pair of vertices being connected with probability p .

Table 2. Clustering coefficient comparison

Snapshot date	Gnutella	$G(N,p)$
2003.11.20	0.008336	0.000179
2004.01.08	0.007731	0.000147
2004.01.25	0.009075	0.000153
2004.02.16	0.007137	0.000152
2004.03.25	0.007019	0.000161

Table 3. Characteristic path length comparison

Snapshot date	Gnutella	$G(N,p)$
2003.11.20	3.35	4.01
2004.01.08	3.64	4.25
2004.01.25	3.77	4.40
2004.02.16	3.49	4.11
2004.03.25	3.37	3.93

3.2 Power-Laws Modeling

The major limitation of the described small-world models is due to increasing evidence of various power-laws of the form $y = x^a$, governing distribution of various graph metrics for many large, self-organizing networks. Faloutsos et al [11] discovered four of these power-laws characterizing topology of the Internet at both inter-domain and router level. These power-laws are defined as follows:

Power-Law 1. (rank exponent R): The outdegree, d_v , of a node v , is proportional to the rank of the node, r_v , to the power of a constant, $R: d_v \propto r_v^R$. The rank r_v of a node, v , is defined as its index in the order of decreasing outdegree.

Power-Law 2. (out-degree exponent O): The frequency, f_d , of an out-degree, d , is proportional to the out-degree to the power of a constant, $O: f_d \propto d^O$.

Power-Law 3. (hop-plot exponent H): The total number of pairs of nodes, $P(h)$, within h hops, is proportional to the number of hops to the power of a constant, $H: P(h) \propto h^H, h \ll \delta$, the diameter. The number of pairs $P(h)$ is the total number of pairs of nodes within less or equal to h hops, including self-pairs, and counting all other pairs twice.

Power-Law 4. (eigen exponent E): The eigenvalues, λ_i of a graph are proportional to the order, i , to the power of a constant, $E: \lambda_i \propto i^E$.

Several research groups have also independently discovered evidence of the same power-laws describing structural properties of the web graph. Since these discoveries occurred on various scales and levels of granularity, they could be taken as indications of possible self-similar or fractal nature of the web. This observation led the authors in [11] to suggest the use of power-law exponents as a way of characterizing different families of graphs. In addition, they demonstrated how these exponents could be used to approximate important graph metrics, such as the number of nodes, the number of edges, the average neighborhood size, and the effective diameter. The significance of these power-laws is that they clearly outline the inadequacy of the described small-world models to accurately capture the true nature of many large networks.

Upon analyzing the Gnutella topology data obtained using our network crawler, the paper discovers it obeys all four of the power-laws described in the previous section. Power-laws relationships between variables are typically plotted on a logarithmic scale, since their plot should, by definition, appear linear. Power-law exponents can then be defined as the slope of this linear plot. We used linear regression to fit a line in a set of two-dimensional points using the least-square errors method. To quantify the validity of the approximation, with each figure we included the absolute value of the correlation coefficient r ranging between -1 and 1 . A $|r|$ value of 1 indicated perfect linear correlation. As mentioned earlier, power-law 1 is evaluated by sorting all nodes in descending order according to their degree, and plotting degree versus rank of a node in this sequence on a log-log scale. The measured data is represented by points $+$, while the solid line represents the least-squares approximation. Figure 2. (a) shows this power-law 1 holds for the Gnutella topology 2003.11.20 sampling instance with rank exponent $R=-0.4741$ and the correlation coefficient of 0.9874 . All correlation coefficients are higher than 0.97 in five samplings. The paper plots the frequency f_d versus the outdegree d of 2004.01.25 sampling instance in log-log scale in figure 2. (b), and the solid line are the result of the linear regression. The result show that the instance with out-degree exponent $O=-2.4866$, the correlation coefficient of 0.9811 . All correlation coefficients are between $0.9698-0.9811$. Figure 2. (c) and (d) show power-law 3

and 4 hold for the Gnutella topology sampling instances 2004.01.08 and 2004.02.16, with hop-plot exponent $H = 5.6789$, the correlation coefficient of 0.9937 and eigen exponent $E = -0.3521$, the correlation coefficient of 0.9858, respectively. All correlation coefficients are between 0.9891-0.9937 and 0.9769-0.9858 in five samplings, respectively.

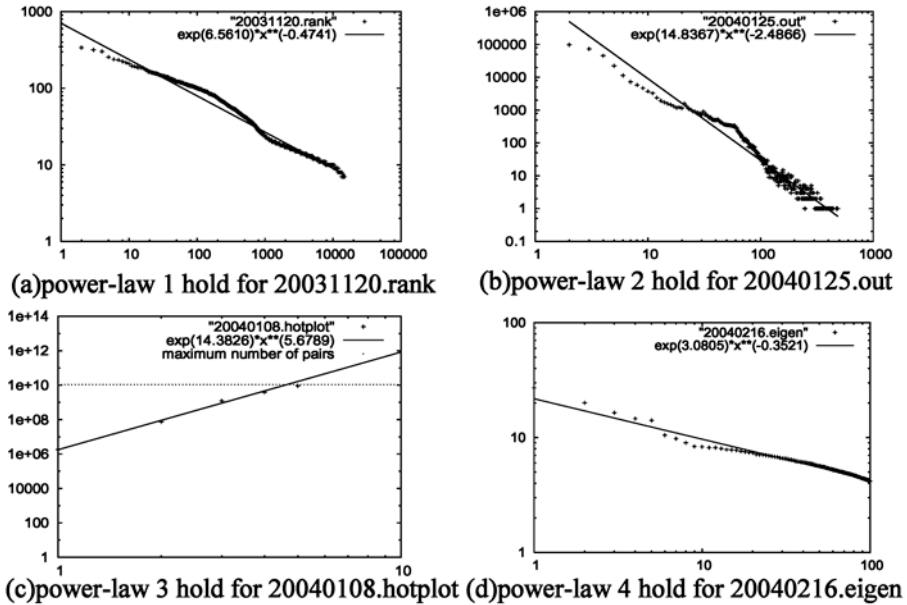


Fig. 2. The four power-laws plots

Our empirical results clearly outline strong power-law properties on the Gnutella network topology. It is our paper that these properties can be utilized to improve performance of algorithms such as those used for searching. In addition, we believe that an accurate model of the network topology of P2P network applications such as Gnutella must inevitably exhibit presence of power-laws 1 and 2, as well as produce all four power-law exponents in close agreement with the ones observed empirically.

4 Conclusions

The paper's main contribution is a novel way to study P2P network topology, namely through small-world and power-laws. The small-world network modeling shows that characteristic path length of P2P network topology is comparable to that of a random graph, while the clustering coefficient is considerably higher. These power-laws capture concisely the highly skewed distributions of the graph properties and quantify them by single number, the power-law exponents. Modeling of P2P network can provide insight for the designers of P2P networks and protocols into the nature of underlying network, help them in understanding of related network structures, facilitates design of new scalable algorithms, allows generation of realistic topologies for simulation purposes and prediction of future trends.

References

1. Napster, <http://www.napster.com>.
2. Freenet, <http://freenet.sourceforge.net>.
3. Gnutella, <http://rfc-gnutella.sourceforge.net>.
4. IMesh, <http://www.imesh.com>.
5. EDonkey, <http://www.edonkey2000.com>.
6. Bittorrent, <http://bitconjurer.org/BitTorrent>.
7. KaZaA, <http://www.kazaa.com>.
8. Morpheus, <http://www.musiccity.com>.
9. Bearshare, <http://www.bearshare.com>.
10. Limewire, <http://www.limewire.com>.
11. Michalis Faloutsos and Petros Faloutsos and Christos Faloutsos: On Power-law Relationships of the Internet Topology, In ACM SIGCOMM '99: 251-262, 1999
12. Shubho Sen and Jia Wang: Analyzing Peer-to-Peer Traffic Across Large Networks, In ACM/IEEE Transactions on Networking, Vol. 12, No. 2, April 2004
13. M. Jovanovic, F. S. Annexstein, and K. A. Berman: "Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella," In Technical Report, University of Cincinnati, 2001
14. B. Krishnamurthy, J. Wang, and Y. Xie: "Early Measurements of a Cluster-based Architecture for P2P Systems," In Proceedings of ACM SIGCOMM Internet Measurement Workshop, November 2001
15. M. Ripeanu and I. Foster, Mapping the Gnutella Network, In IEEE Internet Computing, January 2002
16. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. In Nature, 393:440-442, June 1998
17. Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo. A quantitative comparison of graph-based models for Internet topology. In IEEE/ACM Transactions on Networking, 5(6):770-783, December 1997

A Virtual Tape System Based on Storage Area Networks

Fei Mu, Ji-wu Shu, Bigang Li, and Weimin Zheng

Tsinghua University, Beijing, China
`mufei@mails.tsinghua.edu.cn`

Abstract. The access speed of tape has become one of the main bottlenecks in backup work, because tape backup systems are used quite widely. This paper describes the design and implementation of a Virtual Tape System (VTS) based on the Storage Area Network (SAN), and proposes a tape virtualization technology implemented on the SCSI command level. The system controls the SCSI command stream transferred in the SAN precisely. By transforming the SCSI sequential commands for the tape device into the SCSI block commands for the cache disk, the system makes the cache disk appear and function just like a traditional tape library. Hence, the VTS speeds up the backup process, and reduces the backup window. In addition, the VTS is transparent to users, and has broad compatibility with multiple operating systems and various kinds of backup software. When adopting the virtual tape as the primary backup device, the performance lost is less than 2% compared to backups that use the physical disk as the primary backup device. Another testing result showed that the VTS meets the requirements of backup under a SAN environment and supports multiple backup software and operating systems.

1 Introduction

In this information age, data has become the most important element of an enterprise. To protect file systems from user errors, disk failures, software errors that may corrupt the file system and natural disasters, backup is the most popular solution [1]. The amount of the backup data is always quite large. Tapes have a lower price than the disks with the same capacity [2], so they are widely used for backing up data. Nowadays, about 90 percent of global digital information is stored on tape devices; only 10 percent of those are stored on disks. According to Millward Brown IntelliQuest's report, 85 percent of companies use tapes as their primary backup devices.

As the amount of backup data grows rapidly, the speed of the tape drive becomes a distinct bottleneck in the flow of backup data. If a user wants to back up 1 GB of data, it will take about 2 to 5 hours, which is a terrible waste of time for a busy company. As the capacities of new magnetic disk drives continue to increase at a high rate [3], some companies have begun to use disks as their primary backup devices. The backup data are stored on the disks first, and then

the backup administrator will control the data to be transferred to tape libraries later[4]. The process is apparently complex and involves human actions, which may cause user errors and lower efficiency.

Virtual tape technology makes the disks appear and function just like a traditional tape library, which makes it greatly convenient for users. With even the slowest disk subsystem offering higher throughput than most tape backups, backup time is cut. Further, since all current data resides on the local disk device, restoration can be performed without the need to retrieve offsite tapes, further reducing the time required to restore data[5].

Virtual tape systems can be either hardware- or software-based. Mirage Virtual Tape Controller (VTC)[6] is a hardware-based product that connects servers, disk storage and tape storage together. Seen by the system application, it provides a conventional tape library with high performance backup and restoration, and instantaneous file access capability along with seamless scalability. It supports all major backup software and operating systems. But since it is a hardware-based system, it requires proprietary hardware and has severe limitations in flexibility, throughput, and scalability.

BrightStor CA-Vtape Virtual Tape System[7] is a fully functional, completely software-based virtual tape system. Instead of using a tape, it compresses the data, and writes it to a virtual volume. It cannot support multiple operating systems.

This paper describes the design and implementation of a software-based VTS in a SAN[8][9] environment. The SAN uses extensible network topology to implement centralized data management within a specialized local area network. Ashish and his group implemented a SCSI target system for SANs[10]. This target module was running in the Linux Kernel and was named the SCSI Target Middle Level (STML). It controlled the disk resources and shared them with the FC (or IP) networks. This target driver receives SCSI commands from the network driver and sends these commands to the `sd_mod`. After these commands are executed, STML would catch the responses and send them to the network driver.

Based on the STML, by transforming the SCSI sequential commands for the tape device into the SCSI block commands for the cache disk, the VTS makes the cache disk appear and function just like a traditional tape library. And the VTS transfers data to the real tape resource automatically when the system is in a low-load status. The VTS is not dedicated to any backup software, and it has broad compatibility with multi operating systems and with various tape resources.

2 Architectures

As shown in figure 1, the virtual tape system is made up of six main components: the *SCSI command analysis module*, the *SCSI command transform module*, the *LBA mapping module*, the *virtual tape control data*, the *data transfer module* and *user configuration tools*. The *user configuration tools* lies in the user's space; the other five lie in the kernel space.

The SCSI command analysis module receives SCSI sequential commands from the application servers, determines whether the commands should be executed on the virtual tape or on the physical tape, and delivers them to the proper device. The SCSI command transform module is responsible for transforming SCSI sequential commands into SCSI block commands. This module works with virtual tape control data and LBA mapping module together. The virtual tape control data contains the key parameters of the VTS, such as the virtual tape's size and the space allocation of the cache disk. The LBA mapping module maintains the mapping information between the logical unit of the virtual tape and the logical block address of the cache disk. With the cooperation of the three modules, the SCSI sequential commands can be transformed into SCSI block commands and delivered to the cache disk. The data transfer module's duty is to write the data on the virtual tape to the physical tape when the system load is lighter or when the free space on the virtual tape is not enough. The user configuration tools lie in the user space, providing an interface for users to set the key parameters of the disk-cache system.

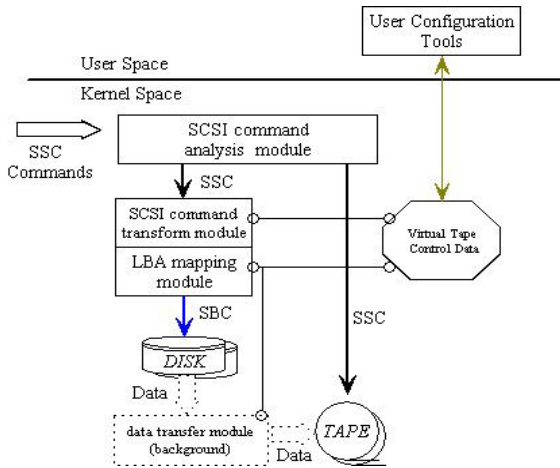


Fig. 1. Architecture of the virtual tape system

3 Implementations

The basic logical unit on the tape is the logical object[11]. For one virtual tape, the VTS maintains a logical object list, which can simulate a tape's behavior and provide convenience in recording commands and mapping logical addresses.

The VTS uses *current logical object* and *current block address* as elementary position pointers. The *current logical object* means the current node in the logical object list. The *current block address* is a logical block address of the cache disk, which indicates the virtual tape's current logical position.

3.1 Logical Objects List

A logical object is either a logical block or a mark. A logical block is a basic unit of data transferred by an application client. A successfully executed write command can generate a logical block on the medium. A mark does not contain user data. In detail, it is either a set mark or a file mark.

A logical objects list is established to record all the logical objects on a virtual tape. One logical object list is attached to one virtual tape. Once a new virtual tape comes into use, a new logical object list is established as well. On flushing the data on the virtual tape to a physical tape, the corresponding logical objects list should be destroyed.

The logical object list is a doubly linked list, which makes it easy to step forward and backward. In the data structure definition, the field *type* indicates which kinds of logical object the node is. The values can be LOGICAL_BLOCK, FILE_MARK, SET_MARK, beginning node and ending node. The field *LBA* is the logical block address on the cache disk at which the virtual tape's logical object is stored. Because the file marks and set marks do not contain user data, they will not be stored on the cache disk. Their LBA fields are useless. The field *size* is the logical object's data length counted in logical blocks of the cache disk.

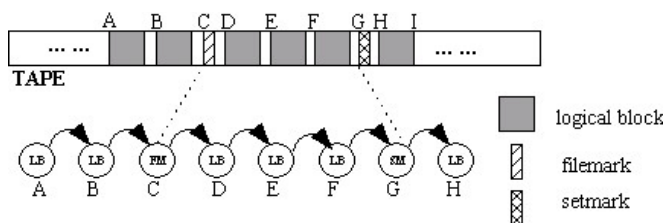


Fig. 2. Logical object list

3.2 Cache Disk Space Allocation

The capacity of a tape library is very large, but that of a single tape is relatively much smaller. So the capacity of a common disk can be larger than that of a tape. The virtual tape system adopts high capacity disks as cache disks. One disk is divided into several sequential-addressed spaces; an individual space can be simulated as a virtual tape. So a disk can simulate multiple tapes, or even multiple tape recorders. The capacity of the virtual tape can be different from that of the physical tape. Figure 3 shows how a disk with the capacity of 80G simulates two tape recorders, and each tape recorder has two virtual tapes with a capacity of 20G.

3.3 Command Transformation

The SCSI sequential command set has important commands whose implementation is mandatory for all SCSI sequential devices. They are INQUIRY, WRITE

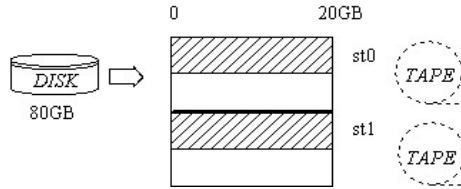


Fig. 3. How a disk simulates tapes

(6), READ (6), WRITE_FILEMARK (6) and SPACE (6). These commands will be discussed in detail.

Commands for Information Inquiring. The virtual tape acts as a real sequential device, which means that the initiator hosts do not know the target devices are disks at all. The initiator hosts acquire the target devices' information through the INQUIRY command [12], so by modifying the information returned by an INQUIRY command we can make the disks appear as tapes.

In the first byte of response data, the lower five bits indicate the PHRIPHERAL DEVICE TYPE. Filling this field with one can make the disk appear as a tape. The VTS should also rewrite the vendor identification, product identification and vendor specific information, which are all contained in the INQUIRY response data.

Commands for Orientation. The SPACE command provides a variety of positioning functions, which are determined by the CODE and COUNT fields. The field of CODE indicates different kinds of marks. Both forward and reverse positioning are provided.

With the logical object list, the SPACE command can be simulated quite easily. We can go over the list from the current logical object in the given direction until reaching the node with a given type for the COUNT times. Then the current logical object becomes the logical object just found, and the current block address becomes the LBA of the found node.

Commands for Data Transferring. If a WRITE (6) command comes when the current logical object node is an ending node, the system will check the command to judge whether it may exceed the boundary of the virtual tape or not. If it may, the system will do nothing but return the information that indicates the virtual tape is full. Otherwise, the system will generate an according SCSI block command. The SCSI block command's logical block address is the current block address. After the command is executed successfully, a new logical object node will be generated, and its LBA is the current block address. Finally, the transfer length will add to the current block address, and the current logical object moves to the ending node.

If a WRITE(6) command comes when the current logical object is a logical block, the system will compare the transfer size with the current logical object's size after executing the generated corresponding SCSI block command. If the two sizes are equal to each other, the logical object list will remain unchanged. Otherwise, the system will change the current logical object's size into the transfer size. If the transfer size is greater than the current logical object's size, the system will seek along the list to find the first logical object whose LBA is larger than the sum of current logical object's LBA and transfer size. This object will be linked to the current logical object and the nodes between them shall be deleted. In the end, the current logical object steps backward and the current block address changes into the new current logical object's LBA.

On receiving a WRITE FILEMARK command the system does not need to generate a SCSI block command. If the current logical object is the ending node, the system will generate a corresponding mark node and insert it in front of the ending node. If the current logical object is a logical block, the system will change its type according to the command and modify its size to zero. The current block address remains unchanged always.

On receiving a READ command, if the current logical object is not a logical block, the system will return failure information. Otherwise, the system can generate a SCSI block command with the current logical object's LBA and size. Finally, the current logical object steps backward and the current block address changes into the new current logical object's LBA.

3.4 Data Movement

By using the logical object list the system can transfer data from a virtual tape to a physical tape. When flushing data, the system will go over the logical object list from the beginning node. When encountering a logical block, a SCSI block READ command will be generated and delivered to the cache disk. The command's logical block address field is the logical object's LBA; its TRANSFER LENGTH is the logical object's size. After reading data from the cache disk, a SCSI sequential WRITE command can be generated, which only needs a transfer length. When a file mark or a set mark is encountered, a WRITE FILEMARK will be generated and delivered to the physical tape. This process is illustrated in figure 4.

4 Testing Results

To test the Virtual tape system, Redhat Linux and Windows 2000 were used in front-end servers. Under the Redhat Linux system, we adopted tar and Taper as backup software; under Windows 2000, we adopted GrBackPro and TH-EasyBackup System. The four kinds of backup software performed well with the VTS. The results indicate that the VTS has compatibility with multiple operating systems and backup software.

We tested the performance of the VTS by using taper under Redhat Linux. The tape drive we adopted was HP C9264CB, and the tape media was HP

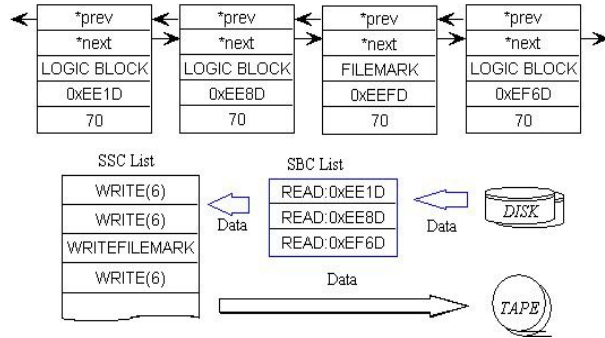


Fig. 4. The process of data flushing

		Backup time for different primary backup devices(s)		
Number of files	Total file size	Physical tape	Virtual tape	Physical disk
12766	58.6MB	174.3	26.5	26.2
1	258.4MB	90.2	7.3	7.2

Fig. 5. The backup time

DLT IV Data Cartridges. We adopted the SEGATE ST3146807FC SCSI disk to simulate a virtual tape. Figure 5 shows the result.

The results showed that the average backup speed of the virtual tape is 98.75% of that of the physical, which indicated that the software cost is quite small. The average backup speed of the virtual tape is 9.5 times of that of the physical tape, which indicated that the VTS can enhance the backup speed greatly.

5 Conclusion

This paper proposes a VTS based on the SAN. The system provides SCSI command reorientation and transformation, which makes the SCSI disks appear and function just like a traditional tape library. The system also provides the ability to move data from a virtual tape to a physical tape automatically. The key feature of the VTS is the logical object list, which can simulate the tapes' behavior easily. It also binds the SCSI sequential command and the transformed SCSI block commands together, which can help move data from the virtual tape to the physical tape conveniently.

Acknowledgement

The works described in this paper are supported by National High-Tech Research and Development Plan of China under Grant (No.2001AA111110). We are grateful to our colleague Luo jun, Meng ran and Zhang Hongcan for their help and encouragement.

References

1. Ann L. Chervenak, Vivekenand Vellanki, and Zachary Kurmas: Protecting file systems: A survey of backup techniques. In Proceedings of the Joint NASA and IEEE Mass Storage Conference, March 1998.
2. Changlong Jiang, Jianping Xiong, Xugang Zhang, Huibo Jia and Duanyi Xu: Sustained Data Recording System Based on Disk Array. IEEE AES Systems Magazine. November 2003.
3. K. Keeton and E. Anderson: A Backup Appliance Composed of High-capacity Disk Drives. HP Laboratories SSP Technical Memo HPL-SSP-2001-3.
4. Kawai H., Sandoh H: An efficient backup warning policy for a hard disk. Computers and Mathematics with Applications, October 2003.
5. Achim Kraiss, Peter Muth, Michael Gillmann: Tape-Disk Join Strategies under Disk Contention. ICDE 1999.
6. Emag Corp.: Mirage virtual tape controller.
<http://www.emaglink.com/MMConnect-VTS.pdf> 2002.
7. CA Corp.: BrightStor CA-Vtape Product Description
http://www3.ca.com/Files/DataSheets/vtape_pd.pdf, 2001.
8. B. Phillips: Have storage area networks come of age? IEEE Computer, vol.31, no.7, 10-12, July 1998.
9. R. Khattar, et al: Introduction to Storage Area Network: Redbooks Publications (IBM), 1999.
10. Ashish P: Design and Implementation of a Linux SCSI Target for Storage Area Networks. Proceedings of the 5th Annual Linux Showcase & Conference, 2001.
11. ANSI, SCSI Stream Commands-2 (SSC-2), <http://www.t10.org>, Sep. 2002.
12. ANSI, SCSI Primary Commands-3 (SPC-3), <http://www.t10.org>, Sep. 2002.

Cocktail Search in Unstructured P2P Networks

Xiuguo Bao¹, Binxing Fang², and Mingzeng Hu¹

¹Research Center of Network and Security Technology
Harbin Institute of Technology, Harbin, 150001, China
{bxg, mzhu}@pact518.edu.cn

²National Network Security Management Center
Beijing, 10029, China
bxfang@pact518.hit.edu.cn

Abstract. Resources search has become a very important area in light of the proliferation of peer-to-peer (P2P) networks and Grids, which heavily rely on effectiveness of forwarding algorithms to propagate queries among peers. Since none of any forwarding algorithms is perfect, Most existed search techniques exhibit varying performances when applying homogeneous forwarding logic in each peer. This paper proposes a promising approach to improve search through heterogeneous forwarding algorithms mixing, named *Cocktail Search (CS)* technique. Its core principle is that available forwarding algorithms are selected probabilistically in each peer to avoid the drawback of single algorithm. In this paper, the basic principle of CS technique is presented and Game theory is employed to verify its feasibility. The simulation results demonstrate that this technique shows significant advantages over others.

1 Introduction

Resource search or discovery is a fundamental issue for Grid and P2P studies, which are both concerned with the pooling and coordinated use of resources within distributed communities and are constructed as overlay structures that operate largely independently of institutional relationships [1]. Search objects may be cycles, storage spaces, files, services, addresses, etc, we generally call them *resources*. Because most popular P2P applications operate on unstructured P2P networks [2-3], and Grids are essentially P2P systems [4], we argue that one shared challenge is how to locate these resources in unstructured networks which is the focus of this paper. In most of search techniques given in the related literatures before this paper, all peers run the uniform processing logic, we call them *homogeneous search (HMS)* techniques. Although various efficient and scalable HMS techniques have been designed [5-9], previous studies showed that the performance of any one single technique is not perfect on all sides [10-11]. Why not mixing those techniques and increasing performance? That is the starting point of our research. We name the search technique based on mixing existed techniques as *Cocktail Search (CS)*, or comparative *heterogeneous search (HES)*. This idea is analogous to the well-known *AIDS cocktail* treatment in our ordinary life.

The goal of this paper is to explore the CS feasibility in the context of the time-keeping P2P networks originated from our prior studies [12]. Usually, a Timekeeping

Overlay Network (TON) consists of continuously running processes inside of each node to make local clocks synchronized with a standard time base like GPS, and these processes are often called NTP processes. We have proposed a *self-organizing* approach for the open problem NTP *autonomous configuration* [13], whose basic idea is that a NTP process should be capable to change its parameters adaptively. In there, the TON is viewed as an unstructured P2P network, and the key stage of autonomous configuration procedure includes configuration parameters search in the TON before fine-tuning those parameters.

2 Principle

HMS techniques in unstructured P2P networks can be categorized as either *blind* or *informed* [8]. The essential difference among various search techniques is either the perspective of query messages forwarding algorithms or propagation rules in a peer. Many forwarding algorithms can be employed, characterized by the number of neighbors of a peer to which a request message is sent, and the way in which these neighbors are selected. Followings are some of them: (1) **Flooding**, use a Breadth-First Traversal for object discovery with *depth or radius* limit D , where D is the maximum of the Time-To-Live (TTL) field in a query measured in hops. (2) **Random-walkers** or gossip, rumor mongering [6,8], forward a query message to the stochastically chosen neighbors at each step until the object is found. (3) **Informed Search**, choose “good” neighbors to forward the query and to reduce overhead with various indices which are related with resources locations. (4) **Iterative Deepening** or expanding ring, gradually enlarge search scope until objects are found or given up. Usually, it is combined with other forwarding algorithms (for instance, Flooding).

What is the most essential property of the resource discovery in unstructured P2P networks? We argue that it is the randomness virtually. Frequently, before searching, we don’t have any knowledge where are the targets, whether they exist, and don’t know which technique is the most suitable. This implies that there is no search technique applicable to all situations, and it may be intelligent and rational if we combine multiple search techniques among peers in some way under these situations.

Based on above observations, the internal logical structure of a peer employing *CS* technique is illustrated in Figure 1. We call the implementation of any *CS* characterized with its forwarding algorithm in a peer as a *module*. Q indicates a coming request message or a new query originated by this peer. The block *MODULES* is a pool of modules available in a peer. The block *Red-killer* is used to check out and then delete redundant or worthless queries, and the block *Check&answer* is for parsing Q with the resources pool block *RESOURCES* whether this peer has the required resources. When a peer forwards Q , the block *MODULE-selector* takes the duty to determine which module will be employed from the block *MODULES*. The block *Forwarder* will execute the *hit* module to propagate Q (the solid lines and dotted lines indicates possible neighbors under using different modules).

The core part of our *CS* search technique depends on intelligently switching existed search techniques or corresponding forwarding algorithms whose function is completed in block *MODULE-selector*. Let’s see how a peer switches its modules.

Suppose $M=\{m^1, m^2, \dots, m^k\}$ denotes the modules set available in the *MODULES* of a peer Γ . Logically, the block *MODULE-selector* is a switch function θ . After a new request comes in, the function θ determines which module will be used for forwarding the incoming request message. However, the function θ does not switch a module deterministically but probabilistically. Because of this, θ is called as a *probability switch* function in this paper. The basic procedure is as follows. When the block *MODULE-selector* receives query Q , a random number X in $[1, k]$ is generated by the function θ that corresponds to the module superscript in M , then this block calls the block *Forwarder* with the parameter X to finish forwarding task.

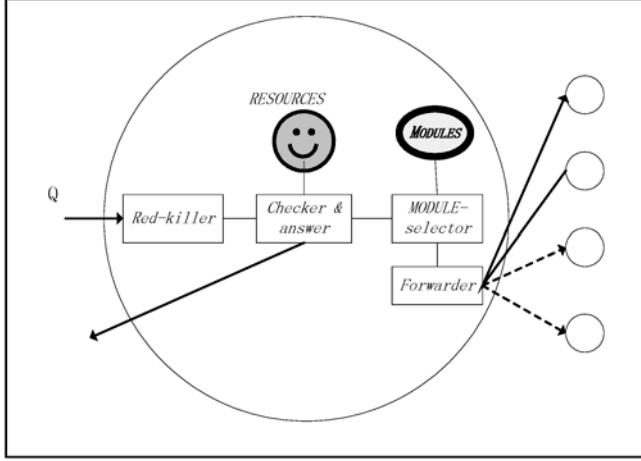


Fig. 1. Peer internal logical structure, and reply message is omitted.

Obviously, the key item is how to construct all *probability switch* functions in all peers to make the overall search performance improved as possible as we can. There are many ways to be imaged out depending on different situations and objectives. This paper only use a simple probability switch function, named the Linear Probability Switch (LPS) as follows.

Let $\Omega = \{1, 2, \dots, k\}$ denote the sample space of random variable X , which is module superscript in set M . If each preference probability of modules in Γ is available, that is

$$LSV = [p_1, p_2, \dots, p_k], \quad p_i \text{ is the preference of the } i^{\text{th}} \text{ module, and } p_i \geq 0, \sum_{i=1}^k p_i = 1.$$

The vector LSV is called as the *linear switch vector* of the peer Γ . Then, we take the probability density function of variable X as

$$f(x) = P(X = x) = p_x \quad x \in \Omega$$

The cumulative distribution function of variable X as

$$F(x) = P(X \leq x) = \sum_{i=1}^x f(i) = \sum_{i=1}^x p_i \quad x \in \Omega \quad (1)$$

So, in this case, the probability switch θ can be achieved by generating random number sequence followed the function F in (1). Now, we can image that the overall search performance of CS technique must emerge changing characteristics with different LSV combinations in peers. An intuitive question arises naturally: does the CS is better than a HMS ?

3 CS Game Model

To verify CS property, we use the game theory. Game theory describes mathematical models of conflicting and cooperative interactions between rational, utility maximizing, decision makers. We found that the searching activities with CS in unstructured P2P networks may be viewed as a non-cooperative, n-player game.

In this game, a peer is a player, which has some resources like files or cycles that other peers may try to discover efficiently, at the same time, it need to search the locations of the required resources as well. Here, We assume each peer utilizes the CS technique to deal with coming queries and its own requests as well, and each player wants to improve search performance. Obviously, the search success rate and the search cost are the most important things that any peer cares for. We argue that the overall search performance will be optimized, when each peer endeavors to get the highest search success rate at the lowest cost. One alternative way to optimize decision-making is cooperating among peers. However, this need very complex decision model and creates more extra communications. Therefore, we assume that peers make these decisions independently or non-cooperatively in the game theory sense. Further, we assume readers have basics to game theory, the formal CS game model as follows.

Let $N = \{1, 2, \dots, n\}$ denote the player set or peer set in an unstructured P2P network. Let A^i denote the pure strategy set of player i that is the available *modules* set, here $1 \leq i \leq n$. We denote

$$W^i = \{p = \{p_1^1, p_2^1, \dots, p_m^1\} : p_j^i \geq 0, \sum_{j=1}^m p_j^i = 1\}$$

It is the set of player i 's mixed strategies over A^i , where we assume $|A^i| = m, m \geq 1$. A mixed strategy p is a probability distribution over player's pure strategies, and which is the LSV in the section 2. Let $W = W^1 \times W^2 \times \dots \times W^n$ denote the set of all mixed strategy combinations. Let $u^i : W \rightarrow [0, 1]$ denote the player i utility or payoff function, and it is some complex function about the search success rate and the search cost according to search performance aims. Let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ denote any mixed strategy profile, where $\alpha \in W, \alpha_i \in W^i, 1 \leq i \leq n$. Consequently, the tuple $G = (N, \{A^i\}, \{W^i\}, \{u^i\})$ is called as the CS game model. We have following very important theorem.

Theorem. *The CS is the best search technique to any HMS when all peers take Nash equilibrium strategy $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$.*

Proof. Firstly, we review the definition of the term Nash equilibrium. A strategy profile $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$ is a Nash equilibrium if no player can gain by unilaterally deviating. We denote $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*) = (\alpha_i^*, \alpha_{-i}^*)$, where α_i^* is a strategy of player i , and α_{-i}^* indicates all other strategies in α^* except α_i^* .

Since our game model G belongs to a non-cooperative, finite pure strategies, and mixed strategies game, according to Nash theorem [14], there must exist at least one strategy profile α^* , for any player i and any of her strategies $\alpha_i \neq \alpha_i^*$, the following must be satisfied is

$$u_i(\alpha_i^*, \alpha_{-i}^*) \geq u_i(\alpha_i, \alpha_{-i}^*) \quad (2)$$

Now let's go to any *HMS*. An employed *HMS* search technique can be looked as a special case of above *CS* game model G . Suppose a module $\hat{m}$ is applied in all peers in this case, and suppose the subscripts of $\hat{m}$ in $A^1, A^2, \dots, A^n$ take 1 wholly. We define a special mixed strategy as follows: $\forall i, W^i = \beta_i = \{1, 0, \dots, 0\}, i = 1, 2, \dots, n$, and the corresponding mixed strategy profile is denoted as $\alpha = (\beta_1, \beta_2, \dots, \beta_n) = (\beta_i, \beta_{-i})$. Since the preference probability of the module $\hat{m}$ is set at 1 and others are set at 0, then the forwarding actions in all peers will uniformly select the corresponding technique of the $\hat{m}$. Therefore, the strategy profile (β_i, β_{-i}) corresponds to a *HMS* technique. According to inequality (2), we have

$$u_i(\alpha_i^*, \alpha_{-i}^*) \geq u_i(\beta_i, \alpha_{-i}^*) \geq u_i(\beta_i, \beta_{-i}) \quad 1 \leq i \leq n \quad (3)$$

Then, the average overall utility functions must satisfy

$$\frac{\sum_{i=1}^n u_i(\alpha_i^*, \alpha_{-i}^*)}{n} \geq \frac{\sum_{i=1}^n u_i(\beta_i, \beta_{-i})}{n} \quad (4)$$

According to inequality (3) and (4), we can conclude: when all peers take Nash equilibrium strategy $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$, the *CS* has the best performance index compared to any *HMS*.

4 Evaluation

Unlike doing evaluation on the generated static topologies [8,10,11], in this paper, we study some more complex retrieving scenarios specially derived from our prior time-keeping project [12]. Our queries for evaluation were selected from TON formation phase that is the core stage for solving NTP *autonomous configuration* issue. A TON likes an *ad hoc* network during construction phase. Before a new lonely node joins, it must firstly find out the desirable addresses (that is the nodes IP addresses) from existed nodes in TON to plug in. Apparently, these IP addresses are *resources* that are peers look for in search technology sense.

What is the determination condition during locating these resources? Usually, each node in TON has a special variable *stratum* according to NTP protocol. For simplic-

ity, the determination condition in our queries is only based on the parameter *stratum expectation (SE)* that means the expected stratum value of a node to find out from TON. For instance, a database server may require *SE* as 2 because its clock precision is very important, and an ordinary workstation may require *SE* as 12. In practical situations, the probability distribution of variable *SE* depends on the usage patterns of joining nodes, but here, we assume it follows the *Gaussian distribution* among 2 and 15. If there are N new nodes, our queries for evaluating *HES* is selected as

Find out addresses to join the TON for N new nodes concurrently, whose determination condition is that the stratum of any node exactly equals to SE.

Three basic *HMS* techniques were implemented for our evaluation: Flooding (FLD), Random-walkers (RDW) and a special informed search technique called as Intelligent Forward (IFD). In our RDW, the query failure is determined by the TTL-based method, and it randomly selects two neighbors to forward requests if possible, i.e. 2-walkers. The IFD module utilizes the neighbor nodes stratum information available in each peer. In IFD technique, requests in any peer will only be passed to neighbors if the *difference* between a neighbor stratum and *SE* is less than the difference between the stratum of current peer and *SE*. Our *CS* is based on above three modules: FLD, RDW and IFD, and assume these modules are available in every peer, and all peers take same *LSV*. In addition, the iterative deepening technique is combined with all tests. We have constructed TON networks in same conditions but with different search techniques, and then observe how many new nodes succeed to join and the corresponding cost. The metrics are counted on the statistics after N nodes are tried to join.

Success rate per node (SR): Rate of successful joined nodes number to N .

Cost ratio per node (CR): Ratio of number of messages processed per node to one's under using FLD technique.

To help our comparison, we take the search cost under using the Flooding technique as the baseline of the metric *CR*.

We use the simulation platform Parsec [15] to implement our experiments. Our main experiment steps are: (1) create 3 root nodes and another 200 nodes as the initial topology of the TON; (2) simulate joining procedure of 8000 new nodes into TON with above 4 different search techniques (3 *HMS* techniques and our *CS* technique) respectively; (3) collect data and calculate above metrics *SR* and *CR*.

The Figure 2(a) gives the performances comparison among four search techniques, and all of them are combined with the iterative deepening technique (solid curves correspond to metric *SR*, and dotted curves correspond to metric *CR*). Here, utilizing our *CS* technique, the curves signed with *HES* takes $LSV=[0.1,0.6,0.3]$ over FLD, RDW and IFD respectively. Especially, in the case of *radius D* equals to 0, that is, none forwarding actions are triggered, repeated experiments show that the metric *SR* during TON construction is about 0.40 in this situation. In Figure 2(a), we can see that the metric *SR* under using *CS* technique is very close to FLD, but with significantly low *CR*. From this, we can get several implications as follows. Firstly, as we expect, the *CS* technique has the power to improve search performance through mixing several *HMS* techniques. Secondly, although the performance under using intelligent forwarding technique is similar to *CS*, its efficiency tightly depends on the resources location information, here it is neighbor stratum. According to above *LSV*, we know that only about *one third* of nodes utilize neighbor stratum information in the curve

HES comparing to IFD, but with similar search performances. In many other situations, it may not possess such exact resources location information like *stratum* in TON. Therefore, this indicates that CS technique may improve search performance without heavily depending on indices like many informed search techniques.

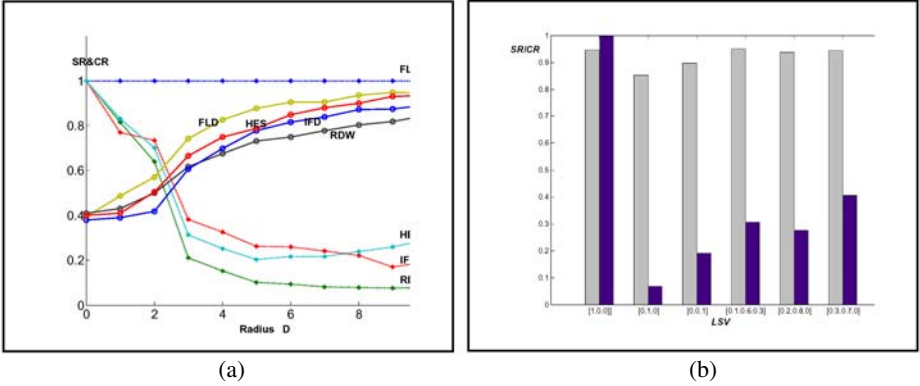


Fig. 2. Comparing about four search techniques, and all combine with iterative deepening technique at uniform expanding pace 1 starting from 1. (a) Performance index SR and CR to radius D . (b) CS performances under different LSV with $D=10$.

The Figure 2(b) gives the performances comparison under different LSV combinations. We can see that tuning LSV will significantly affect CS performance. Note that when LSV takes the combination $[0,2,0,0]$, that is, without utilizing resources location information – neighbor stratum, we also can obtain desirable search performances. Although Figure 2(b) indicates that the **Random Walkers** technique is the most suitable in this case, however, it is not true in many other situations. Therefore, the most outstanding advantage of our CS is it provides the flexibility to tradeoff search success rate and cost and to avoid drawbacks of using single search technique, but any single HMS does not work this way.

5 Conclusions

In this paper, we explore a new approach to improve resource search performance that lively is named as *cocktail search*. The essential difference in CS compared to HMS is on the forwarding algorithms. CS improves performances through rationally mixing multiple forwarding algorithms to avoid their respective drawbacks, and HMS improves the performance through exploring more efficient forwarding algorithms. The CS feasibility is verified with the Game theory that provides a fundamental research point. The simulation experiments in the context of timekeeping P2P networks proved the capability of the cocktail search or heterogeneous search technique. We argue that the CS is simple, generic, promising to improve search performance in unstructured P2P networks and Grids that is a key technology for Internet development. This research was supported by National Natural Science Foundation of China grant CNSF 60203021.

References

1. I. Foster and A. Iamnitchi: On death, taxes, and the convergence of peer-to-peer and grid computing. In *2nd International Workshop on Peer-to-Peer Systems*, 2003.
2. A. Stavrou, D. Rubenstein and S. Sahu. A lightweight, robust P2P system to handle flash crowds. *Selected Areas in Communications, IEEE Journal*, 22(1):6–17, 2004.
3. Bao Xiu-guo, Hu Ming-zheng and Fang Bin-xing. A method of actively resource publishing based on P2P. *Chinese Mini-micro Systems*, 25(4):526–530, 2004.
4. D. Talia and P. Trunfio. Toward a synergy between P2P and Grids. *IEEE Internet Computing*, 7(4):94–96, 2003.
5. L. A. Adamic, B. Humberman, R. Lukose and A. Puniyani. Search in power law networks. *Physical Review*, 64:46135–46143, 2001.
6. P. Marius and S. Aruna. Cost-effective broadcast for fully decentralized peer-to-peer networks. *Computer Communications*, 26(11):1159–1167, 2003.
7. M. Jan and S. Burkhard. A methodology for the design of distributed search in P2P middleware. *IEEE Network*, 18(1):30–37, 2004.
8. C. Lv, P. Cao, E. Cohen, K. Li and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS*, 2002.
9. D. A. Menasce. Scalable P2P search. *Internet Computing*, 7(2):83–87, 2003.
10. D. Tsoumakos and N. Roussopoulos. A comparison of peer-to-peer search methods. In *Proceedings of the Sixth International Workshop on the Web and Databases*, San Diego, USA, 2003.
11. T. Lin and H. Wang. Search performance analysis in peer-to-peer networks. In *Proceedings of Third Peer-to-Peer Computing International Conference*, pages 204–205, 2003.
12. Bao Xiu-guo, Hu Ming-zheng, Zhang Hong-li and Xu Bing-bing. A self-organizing time-keeping network. *Chinese Journal of China Institute of Communications*, 25(1):150–157, 2004.
13. D. Mills. A brief history of NTP time: confessions of an Internet timekeeper. *ACM Computer Communications Review*, 33(2):9–22, 2003.
14. J. Nash. Noncooperative games. *Math. Ann.*, 54 :286–295, 1951.
15. R. Bagrodia, R. Meyer, M. Takai, Y.A. Chan, X. Zeng, J. Marting and H.Y. Song. Parsec: a parallel simulation environment for complex systems. *Computer*, 31(10):77–85, 1998.

Data I/O Optimization in Storage Systems

Di Wang, Ji-wu Shu, and Meiming Shen

Department of Computer Science and Technology
Tsinghua University, Beijing, China
wangdi97@mails.tsinghua.edu.cn

Abstract. One of the most effective ways to improve the I/O performance of a storage system is to enhance the hard disk's read/write ability. We used an I/O processing node in the storage network to optimize data organization and I/O performance. By analyzing existing algorithms and different requirements for read and write operations, we designed an improved optimizing algorithm to schedule disk I/O requests. It selects the closest request in queue to process first, and uses an EW mechanism to modify write locations. Typically, the algorithm can reduce a disk's average response time by about 15%-17%. This paper also presents an EW stripe and copy algorithm that can improve I/O performance using parallel disk accesses, and enhance reliability by data duplication. With one copy preserved, it can reduce the response time by about 30%.

1 Introduction

The capacity of data resources is growing rapidly at a rate of about 50%-100% every year. Traditional storage architectures can not meet the requirements of growing data storage capacity, and integrated scalable storage systems[1] have become necessary.

In integrated storage systems, it is crucial to improve disk access speed and data reliability. By scheduling optimizations of read/write requests, the I/O performance of storage systems can be greatly enhanced[2]. In addition, parallel disk access can be achieved by data striping technology. And for each stripe, some copies may be preserved on different disks. By this technology, not only can data reliability be improved, but the data access speed can also be increased[3].

The QoS of virtual storage systems[4] is a new research focus in the field of storage technology. In [5], some effective attempts are made to improve service quality of disks. An important way to enhance disk performance is to schedule disk I/O requests for each disk. In [6], some advanced scheduling algorithms were introduced. And the dynamic data distribution[7, 8] in whole systems is another important focus of storage research.

We have established a self-developed storage network - the TH-MSNS[9]. Based on it, we implemented some effective data optimizing mechanisms for the storage system. Section 3 introduces an improved scheduling algorithm for disk I/O requests, which can remarkably reduce the average response time of disk operation. We also propose a stripe and copy mechanism based on EW[10] in

section 4. It can improve I/O performance by parallel disk access, and improve the system's reliability by data duplication.

2 The TH-MSNS Storage Network

The data I/O optimizing mechanisms introduced in this paper were implemented on the TH-MSNS, which is a special SAN system using software rather than the usual hardware to control the I/O processing. The TH-MSNS is based on the FC or IP protocol. It can affiliate storage devices into the network, including disk arrays (FC or SCSI) and tape devices (used for data backup). All SCSI devices are joined to the network via an I/O processing node which is called the Multifunctional Controlling Node (MCN).

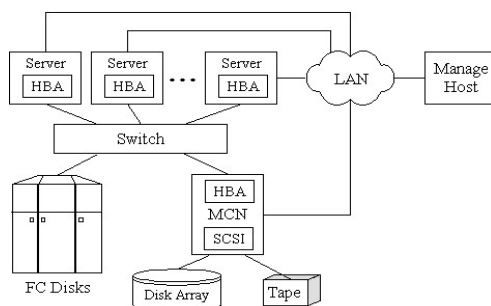


Fig. 1. The Architecture of TH-MSNS

The functions of the SCSI target midlevel layer[11] have been accomplished by software modules on the MCN (with the Linux OS), so that the devices attached to it can be discovered and accessed by front hosts of the storage network. Accordingly, SCSI commands that are issued by the front hosts are first handled by the STML module on the MCN, and then transmitted to the actual disk or tape devices.

By the virtualization implemented on the MCN, we can organize all disk devices to a unified logical space, and provide it to the front hosts. After the address conversion by virtualization, all requests already have physical addresses, and they can be sent directly to the SCSI MOD layer. But in our system, the optimization is accomplished by another type of address mapping, and the addresses are converted to some more rational locations in order to reduce I/O operation time.

In normal storage systems without virtualization, our optimization mechanisms can also be adopted to enhance system performance.

3 Scheduling of Disk I/O Requests

CPUs have a much faster processing speed than disks do. So if we want to improve a storage system's I/O efficiency, one of the most effective ways is to

schedule the requests and allocate data to more rational locations to increase disk accessing speed. In the optimization module, a queue is created for each disk to schedule its requests.

3.1 Popular Scheduling Algorithms

Obviously, the order of a disk's I/O requests can not be changed at will. In the request queue, if 2 requests have their destinations intersected, they might have a logical sequence relationship. But if the two are both read requests, and there is no write request to same destinations, the 2 read requests could be exchanged. Otherwise their orders can not be changed.

One advanced scheduling algorithm is called SATF-EW, which checks the request queue and calculates both seek time and rotational latency of a disk to find the closest request. Some other algorithms take the request deadlines into consideration. Each I/O request must be processed before its deadline. To select the next request, FreeBW-SATF considers both the seek time and rotate latency. While FreeBW-SCAN only calculates the seek time of disk head[6].

In order to prevent the disk head from being trapped in a small region, all the algorithms force the disk head to move in one direction when scheduling, until it cannot move any further. Another common aspect is that all the algorithms use an EW mechanism[10] to optimize write operations. It makes new data be relocated to the nearest free block. The EW can noticeably improve the write speed, but a mapping table is needed for later data access.

3.2 An Improved Scheduling Algorithm

FreeBW algorithms must handle the deadlines for all requests, so their calculations are more complex than SATF-EW. So if the MCN's workload is very heavy, the SATF-EW algorithm should be used to lessen the system's computing burden.

Typically, many requests are sent as sequential read or write requests. If we just consider the factor of the disk seek time, the orders of those sequential requests will seldom be broken. If we also calculate disk rotational latency, a request in the middle of that read/write sequence could possibly be scheduled first. Therefore the continuity of sequential read/write commands would be broken and the performance would decline.

On the other hand, we should consider the disk rotational latency to schedule write requests. Then new data can be relocated to a block which has the minimum seek time and rotation distance from the disk head, and some free blocks can be preserved on that disk track for later use.

Therefore, it is better to use SATF-EW or FreeBW-SATF to process write requests in order to keep free blocks in the disk; but FreeBW-SCAN is preferred for processing read requests to keep their sequence. So here comes a compromise: if there is no write request in the queue, only the seek time is calculated; otherwise, we should consider both seek time and rotational latency. We call the method SATF-EW*(no deadline) or FreeBW-Combined(with deadline).

Another problem is that all these algorithms allow the disk head to move in only one direction at a time. When the read ratio is very high, the EW enhancement is little. If we still restrict the disk head's movement, the performance of local and sequential reads will be impaired. So we not confine the movement of disk head any more at that time.

We propose an improved algorithm to schedule disk requests as follows:

```

relocate new data to the nearest free blocks; /* EW */
if (read ratio is not very high) {
    if (system workload is heavy, and request deadlines are not required) {
        /* schedule with SATF-EW */
        if (there is no write request in the queue) {
            calculate seek distance of disk head for each request;
            select the anterior request with the shortest distance;
        }
        else {
            calculate both seek and rotational latencies for each request;
            select the closest request to process;
        }
    }
    else { /* schedule with FreeBW-Combined */
        determine a deadline for each request;
        if (there is no write request in the queue) { /* FreeBW-SCAN */
            calculate seek distance of disk head for each request;
            select the anterior request with the shortest distance;
        }
        else { /* FreeBW-SATF */
            calculate both seek and rotational latencies for each request;
            select the closest request to process;
        }
    }
}
else
    schedule with FreeBW-SCAN without restriction on disk head movement;

```

This improved algorithm uses EW to write new data to the nearest free blocks. But it requires a large mapping table to be maintained on the MCN; otherwise that data cannot be accessed later. If we can omit the mapping table, the implementation complexity will be reduced enormously. For the algorithm described above, the first step (EW) can be removed, leaving the original write addresses untouched. So it becomes a simplified algorithm which may has a less improvement for processing write requests, but is much easier to implement.

3.3 Experimental Results

We used the Disksim (version 3.0)[12] program to run the simulation experiments. The disk type used in the simulation was a Quantum Atlas10k, and there were 8 disks residing in the storage system. We sent read/write requests

to those disks individually, and the probabilities for different disks were same. The length of the request queue was defined as 8. And we set the deadline for each request to 200 ms after its arrival.

All read/write requests were generated randomly. The request interval was 0-10 milliseconds. The probability of sequential access was 60%, and the probability of near access (within 20,000 blocks range) was 50%. We set the disk utilizing ratios to 0.5. For write requests, the possibilities of updates and new data additions(EW needed) were both 50%.

The performance of different scheduling algorithms is shown in figure 2.

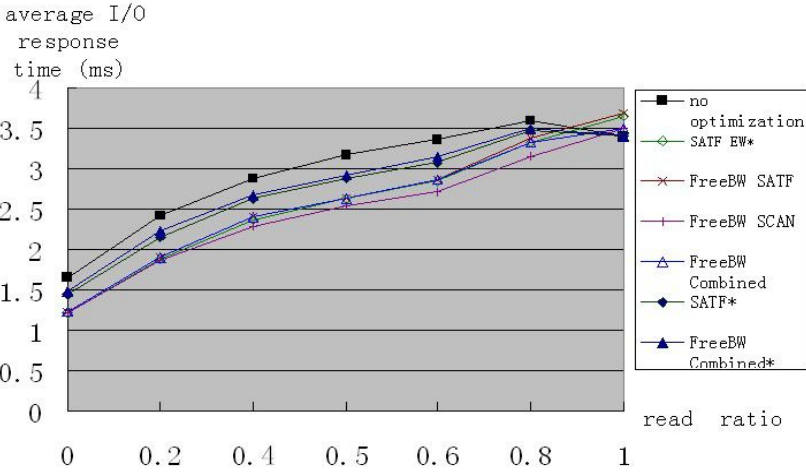


Fig. 2. Algorithm performances with different read ratios

As we can see from the results, when the read ratio was more than 90%, the normal algorithms would lead to a performance reduction. So when the read ratio is more than 90%, the modified FreeBW-SCAN (without disk head movement restriction) should be adopted. If the read ratio is lower than 90%, the improved algorithm performs as SATF-EW* or FreeBW-Combined, as shown in figure 2. When read and write requests are fairly balanced, the algorithms of FreeBW-Combined and SATF-EW* can reduce the average response time of disk I/O operations by 15-17%.

We also tested performance of the simplified scheduling algorithm (without the EW mechanism), and the results are also shown in figure 2. Under different workload conditions, the simplified algorithm utilizes SATF* (SATF-EW* without EW) or FreeBW-Combined* (no EW) to schedule disk requests. According to the results, the simplified algorithm can reduce the disk response time by about 8-10%.

4 Data Striping and Duplication

In storage systems, large amounts of data can be divided into small stripes (similar to RAID 0). So the system I/O performance can be increased by parallel

disk access. On the other hand, copies of data can be saved on different disks to enhance the storage system's data reliability.

4.1 The EW Stripe and Copy Algorithm

In our EW stripe and copy algorithm, large data is divided into several stripes and written onto disks respectively. For each stripe, some copies of it will be made and saved on some different disks. And all disk write operations here use the EW mechanism.

When reading the original data, each stripe will search its copies to find the one closest to current disk head position, and then use that copy to complete the read operation. Hence the read time will be reduced. When a disk is damaged, the original data can still be read from copies saved on other disks.

We should select appropriate disks according to their respective conditions. In our experimental, since the usage ratio of each disk was the same, we chose disks that had the fewest requirements in the previous period to contain data stripes and copies. If their usage were different, the disks that had the greatest amount of free space would be preferred.

The EW stripe and copy algorithm is as follows:

```

for (each requests received){
    divide the data into M stripes according to the stripe size;
    if (the request is a read one){
        for ( M stripes of it){
            search for the copy that is closest to the disk heads among N copies;
            read that copy of data;
        }
    }
    if (the request is a write one){
        select M disks to save stripes;
        for (M stripes){
            write a stripe in that disk;
            for (N copies){
                select a disk from other disks;
                write the data in that disk with EW mechanism;
            }
        }
        update the address mapping table;
    }
}

```

We could make the quantity of data copies alterable: when there is enough free disk space, more data copies can be kept to improve the reliability. But if the disks are relatively full, the number of copies should be reduced to release more space for new data.

4.2 Results and Performance Analysis

Using the same experimental environment described in 3.3, we ran the tests with the interval of requests of 0-50 ms and 0-100 ms. 10,000 read/write requests were

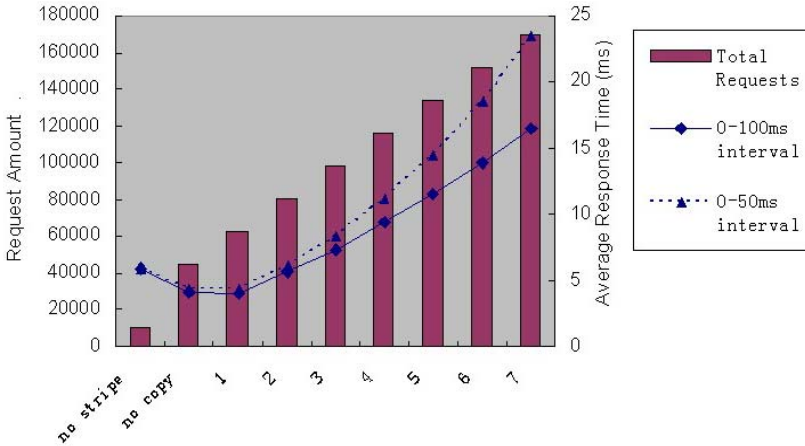


Fig. 3. Performance of EW stripe & copy algorithm with different number of copies

randomly generated with a read ratio of 60%, and the stripe size was 8 blocks. The experimental results are shown in figure 3.

For the x-coordinate, "no stripe" means the condition without striping or copying; "no copy" means only striping was implemented ; and the numbers denote the number of data copies. In figure 3, the histogram presents the request amount, and the dots and lines show the average disk I/O response time.

As the results show, with striping and copying, the total amount of I/O requests increased steeply. If only striping was adopted, a request with large data would be converted to some parallel disk requests with small data, so the average response time could be reduced by about 25%. When the number of copies was 1 or 2, since the read request could select the closest copy to access, the disk response time would also drop. But if the copy number was more than 3, the amount of write requests would increase rapidly, which would cause the requests to be stacked in the operation buffers of the disks. Hence the performance would falls remarkably.

To achieve the best I/O performance, only one copy of each data stripe should be saved. If data reliability is highly demanded, 2 copies can be preserved for each stripe. With the request interval of 0-100 ms, making one copy can reduce response time by 32%, and making two copies can reduce it by 5%. Since the possibility of a 2-disk failure is very low, 2 copies can ensure sufficient reliability. continue increasing the copy number would only lead to an overburden on the storage system and a drop in I/O performance.

When the maximum interval was 50 ms, the results were similar, but responses were slower as the number of copies increased. We also discovered in experiments that if the original requests had a very high frequency(with intervals of 0-10ms), only using the striping operation could almost double the response time. And it would be much longer with data copying. This was caused by the request stacking in the disk operation buffers. Therefore, if the write operations

are over crowded, only one copy should be kept for each stripe. The duplication could even be skipped until the system is free to lessen the system burden.

After the striping and copying disposal, requests for each disk can be scheduled, as described in section 3. Because our stripe and copy algorithm is based on the EW mechanism, the simplified algorithm described in 3.2 could be used directly for further performance optimization.

5 Conclusions

One of the most effective ways to improve a storage system's I/O performance is to enhance the read/write ability of its hard disks. We used an I/O processing node (MCN) in the storage network to schedule the request queue for each disk. By analyzing some existing algorithms and different requirements for read and write operations, we designed an improved optimizing algorithm for scheduling disk I/O requests. In the conditions that read and write requests are balanced, the algorithm can reduce the disk's average response time to I/O operations by 15%-17%. The experiments also showed that if write locations were not modified, the simplified algorithm could still bring a performance improvement of 8%-10%.

Additionally, large amounts of data can be divided into small stripes to improve accessing speed by parallel disk operations. At the same time, making copies for each stripe could increase data reliability. So we proposed an EW stripe and copy algorithm to achieve these goals. Although the algorithm causes a proliferation of I/O requests, it can effectively reduce the disk response time when the number of copies is small, according to experimental results.

Acknowledgement

The work described in this paper was supported by the National High-Tech Research and Development Plan of China under Grant (No.2001AA111110).

References

1. D.Tang. Storage area networking-the network behind the server. Technical Report, Gadzoox Microsystems, 1997.
2. F.I.Popovici, A.C.Arpati-Dusseau, R.H.Arpati-Dusseau,Robust. Portable I/O Scheduling with the Disk Mimic. The 2003 USENIX Annual Technical Conference.
3. M.Uysal, A.Merchant, G.A.Alvarez. Using MEMS-based storage in disk arrays. In the Proceedings of the 2nd USENIX Conference on File and Storage Technologies, March 2003.
4. C.S Kim, G.B Kim, B.J.Shin. Volume Management in SAN Environment, ICPADS 2001:500-508. 1997.
5. L.Huang. Stonehenge: A High Performance Virtualized Storage Cluster With QoS Guarantees. State University of New York at Stony Brook, 2002.

6. C.Zhang, X.Yu, A.Krishnamurthy, R.Y.Wang. Configuring and Scheduling an Eager-Writing Disk Array for a Transaction Processing Workload. FAST 2002: 289-304.
7. A.Brickmann, K.Salzwedel, C.Scheideler. Efficient,distributed data placement strategies for storage area networks. In the proceedings of the 12th ACM Symposium on Parallel Algorithms and Architectures (SPAA), pages 119-128, ACM press, 2000. Extended Abstract.
8. R.J.Honicky, E.L.Miller. A fast algorithm for online placement and reorganization of replicated data. 17th International Parallel and Distributed Processing Symposium (IPDPS), 2003.
9. J.W.Shu, J.Yao, C.D.Fu, W.M.Zheng. Highly efficient FC-SAN based on load stream. The Fifth International Workshop on Advanced Parallel Processing Technologies, LECTURE NOTES IN COMPUTER SCIENCE 2834, pp.31-40,2003, Spring-Verlag Berlin Heidelberg printed.
10. R.M.English, A.A.Stepanov. Loge: a selforganizing storage device. In the Proceedings of USENIX Winter'92 Technical Conference, pages 237-51. USENIX, January 1992.
11. A.A.Palekar, R.D.Russell. Design and Implementation of a Linux SCSI Target for Storage Area Networks. In the Proceedings of the 5th Annual Linux Showcase & Conference, Nov.2001.
12. J.S.Bucy, G.R.Ganger. The Disksim Simulation Environment Version 3.0 Reference Manual. CMC-CS-03-102, Jan.2003.

Data Resource Discovery in a Computational Grid

Sajindra Jayasena¹, Chin-Peng Yee¹, Jie Song², Abele Stoelwinder²,
Chong Wee See³, and Wai-Hong Wong⁴

¹ Singapore-MIT Alliance, National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
sajindra@mit.edu, chinpeng15@hotmail.com

² Nanyang Center of Supercomputing and Visualization, MPE
Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
{asjsong,MAStoelwinder}@ntu.edu.sg

³ Asia Pacific Science and Technology Center, Sun Microsystems Inc.
Simon.see@sun.com

⁴ Singapore Institute of management, Open University Center
wongwh010@hotmail.com

Abstract. A computational grid focuses on the effective sharing of computing resources instead of data resources over the grid. However, distributed or parallel scientific and engineering applications often require wide access to large amounts of data. The demands to serve data are often as important as computational resources. Therefore, our focus is on the discovery of data resources leveraging on existing computational grid management technologies. In this paper, we proposed a data resource discovery mechanism to allow the Sun Grid Engine to retrieve, index and monitor the metadata from Oracle 10g. In the implementation, we analyze the organization of metadata information in Oracle 10g - Oracle's data grid solution, and the resource discovery mechanism in Sun Grid Engine, one of the widely used computational grid middleware. The proposed mechanism provides a unique method to share data resources in the same way as computational resources. This allows the computational grid to be better integrated with the data grid.

1 Introduction

In the computational grid, the computing resources are treated as a utility without being concerned about its location and demographics. However, distributed or parallel scientific and engineering applications often require access to large amounts of data. A data grid can be used to satisfy these demands. This creates two disparate and independent grids, one for computation and the other for data. Another idea would be to combine the two under the same umbrella. This results in the locating and serving of data based on locality, load and the underlying distributed data storage mechanism, and leveraging on existing grid management technology that mainly focus on computational resources.

In this paper we explore the mechanism to integrate data grid resource discovery within the computational-centric grid. We focus on how to extend grid management services of Sun Grid Engine (SGE)[2] – which focuses on computation resource us-

age rather than data resource usage, with Oracle's answer to data grids, Oracle 10g[1]. Figure 1 shows an SGE grid and an Oracle grid working independently to handle computational and data jobs respectively. This would result in less interoperability and results in confusion when managing resources. As an integration mechanism, the Globus [3] toolkit can be used to interface between SGE and Oracle. However, there can be substantial resource overheads by using Globus as an interfacing tool. Fundamentally, a computation resource and a data resource can be treated as a kind of grid resources. Thus they should be able to be managed together by the SGE resource management in some way.

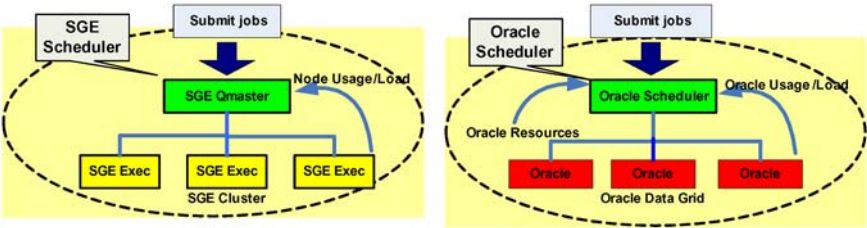


Fig. 1. Division of SGE and Oracle scheduler

Figure 2 depicts the envisioned interfacing of SGE with a data grid. The Oracle nodes would be installed on SGE Execution daemons [7] so that SGE Qmaster daemon [6] could dispatch data grid related requests to those Oracle nodes depending on their resource characteristics, load/usage and that of the job itself.

In this architecture, we intend to extend SGE resource discovery and indexing API to retrieve the metadata information of Oracle instances installed on execution hosts. We also modify the load/usage sensing modules to collect usage status of data resources. In this way, SGE can directly access and monitor the data resource in oracle database without interaction with oracle grid scheduler.

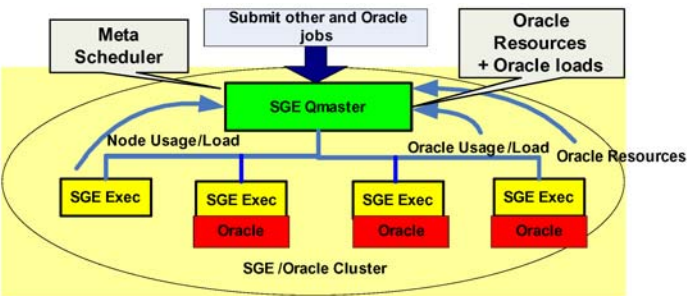


Fig. 2. Integration approach

The remainder of this paper is organized as follows: The Overall architecture is introduced in Section 2. Section 3 describes the three steps of integration: 1) Extracting Meta information from Oracle database; 2) Monitoring Oracle load/usage using SGE load sensors; and 3) Database resource Indexing in the Qmaster. Finally a conclusion is in Section 4.

2 Architecture

In order for SGE to retrieve the data resources, the architecture of integrating Oracle 10g data grid services with the SGE grid management system is depicted in Figure 3.

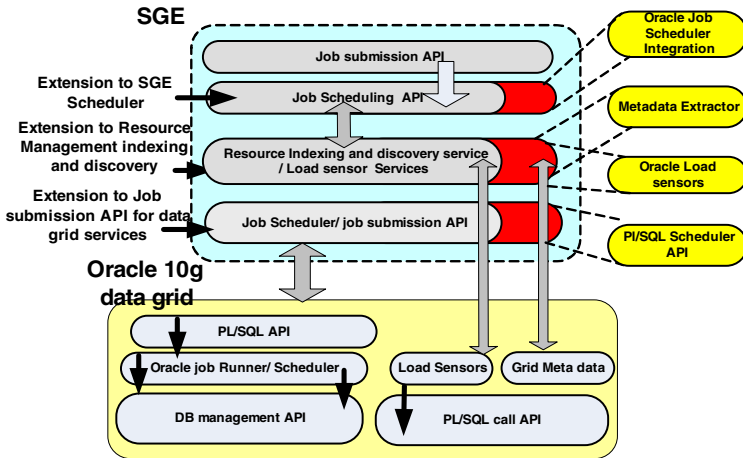


Fig. 3. Integration Architecture

In computational grids, the resource details indexed by grid services are the hardware resource limitation, current usage, application availability, etc. However, in a data grid the concept of resource monitoring goes beyond this. In addition to tracking and resolving resource (table spaces, user access privileges) it needs to know the locations of the schemas and related objects. We intend to add this additional functionality to SGE's resource monitoring and discovery routines to facilitate in Oracle job scheduling. Further there can be Oracle specific load/usage details that would provide more insight than generic host specific load parameters when making the selection decision for an execution node. Then in order to schedule Oracle jobs, we need to incorporate these resources and load information with the SGE Scheduler to make a viable node selection decision. Finally execution nodes need to be incorporated with the logic to submit the jobs to the selected Oracle instance.

SGE distinguishes its nodes into *Master*, *Submit* and *Execution* nodes. Keeping in line with this notation we would label the Oracle database nodes as *execution hosts* where the SGE *Master host* could explore and resolve the resources and database details as well as submit Oracle jobs to the specific execution hosts depending on the usage/load and resource availability.

3 Implementation

In this section, we address the integration features step by step to guide through the usage of the system. It starts from discovering resources of Oracle 10g, monitoring its

load and usage and how they would be integrated with SGE's resource description and load monitoring framework. Then we would explain how SGE would use these load and resource information to schedule data grid jobs, and how they are dispatched and executed by the execution hosts.

3.1 Metadata Extraction from Oracle Instances

First we will look at how SGE get to know about Oracle resources. The current implementation in SGE for resource registration is handled by the SGE Execution daemons. There is no concept of resource discovery in the current SGE implementation. Currently it has the following limitations:

- 1) In most cases the execution hosts are treated as homogeneous nodes where differences in resources are treated insignificant.
- 2) Only primitive resource details such as the existence of software licensed at cluster and host level are handled. It cannot handle multi-valued complex resource hierarchies like multiple database schema resources – tables, procedures, programs etc.

Therefore we had to integrate a resource discovery agent with the execution daemons running on Oracle instance as shown in Figure 4. This JAVA based agent would interface with the schemas and provide the schema metadata like table structures, stored procedures, programs etc. This extends from the SGE abstraction of defining resources – '*complexes*' [4]. Each resource type (tables, procedures, and programs) is treated as a different *complex* by the Scheduler/*Qmaster*, which uses them to schedule jobs by resolving the services published by the execution daemons. This concept is similar to the service discovery and resolving in current grid standards like GRIS in Globus [3]. These complexes can be defined as SGE's global and host specific *consumables* resources [4].

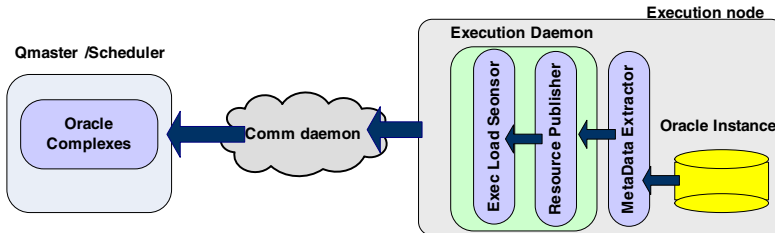


Fig. 4. Metadata extraction integration

The following outlines a set of Metadata extracted from an Oracle instance (namely procedure, table and Oracle program information) that is sent to the *Qmaster* node through the load sensor interface of the underlying execution host named *smal-06.ddns.comp.nus.edu.sg*.

One issue is that, as shown in Figure 6, the lists returned from the load sensor script can be very long if there are many procedures, programs or tables. This resulted in buffer overruns in the *Execution* daemon. To facilitate large streams we incorpo-

rated a batch processing approach where our Oracle Load sensor module breaks up the long resource list to configurable-length sub lists and send as separate lists. Then at the *Qmaster* they are aggregate to a single resource list and indexed in the resource tree.

```

sma1-06.ddns.comp.nus.edu.sg:Oracle.Procedures:
#SYSTEM:SYSTEM.GET_LOADS,SYSTEM.INTERNAL_SURROGATE_SYST
EM,SYSTEM.LOADPROVIDER,SYSTEM.ORA$_SYS_REP_AUTH,SYSTEM.
PRINT_DETAILED_REPORT,SYSTEM.PRINT_RUN,SYSTEM.PRINT_SUM
MARIZED_REPORT,SYSTEM.PRINT_UNIT,SYSTEM.ROLLUP_ALL_RUNS
,SYSTEM.ROLLUP_RUN,SYSTEM.ROLLUP_UNIT,SYSTEM.SET_WINDOW
_SIZE,SYSTEM.TESTSAJI,SYSTEM.TESTSAJI2

sma1-06.ddns.comp.nus.edu.sg:Oracle.Programs:
#SYSTEM:SYSTEM.TESTSAJIPROGRAM,SYSTEM.TESTPROGRAM4,SYST
EM.TESTPROGRAM,SYSTEM.TESTPROGRAM3,SYSTEM.TESTPROGRAM2

sma1-06.ddns.comp.nus.edu.sg:Oracle.Tables:
#SCOTT:SCOTT.BONUS,SCOTT.DEPT,SCOTT.EMP,SCOTT.SALGRADE

```

Fig. 5. Metadata extracted by the load sensor

```

SMA1-06.ddns.comp.nus.edu.sg:Oracle.Tables:#SYSTEM:SY
STEM.AQ$_INTERNET_AGENTS,SYSTEM.AQ$_INTERNET_AGENT_PRIV
S,SYSTEM.AQ$_QUEUES,SYSTEM.AQ$_QUEUE_TABLES,SYSTEM.AQ$_
SCHEDULES.....

SMA1-06.ddns.comp.nus.edu.sg:Oracle.Tables:#SYSTEM:SY
STEM.DEF$_DESTINATION,SYSTEM.DEF$ERROR,SYSTEM.DEF$_LOB,
SYSTEM.DEF$_ORIGIN,SYSTEM.DEF$_PROPAGATOR,SYSTEM.DEF$_P
USHED_TRANSACTIONS....

SMA1-06.ddns.comp.nus.edu.sg:Oracle.Tables:#SYSTEM:SY
STEM.HELP,SYSTEM.LOGMNRC_DBNAME_UID_MAP,SYSTEM.LOGMNRC_
GSII,SYSTEM.LOGMNRC_GTCS,SYSTEM.LOGMNRC_GTLO...

```

Fig. 6. Batched resource details propagation

3.2 Oracle Load Sensors for Oracle Load /Usage Monitoring

After discovering the resources, the next step is to measure the database loads periodically. The SGE default load sensors [5] only provide a generic set of load information, such as available disk space, virtual Memory, CPU usage etc. Therefore, in order to make a suitable scheduling decision for an Oracle job, the SGE Scheduler needs to know the Oracle specific loads in addition to the host and SGE queue specific loads. As shown in Figure 8, we implemented a JAVA based load information extractor that would periodically propagate the Oracle load details to SGE's load

sensors. Our Oracle load sensors integrated with SGE execution node load sensors provide the following additional information.

- 1) Concurrent connected sessions/users.
- 2) Number of concurrent Database reads /writes.
- 3) Size of allocated buffer spaces in Oracle heaps in Megabytes.

These load parameters are propagated to Scheduler and *Qmaster* daemons through the SGE's load sensor infrastructure. The Execution hosts register these new load parameters as *host specific Complexes* [4]. A sample load sensor feed from an execution node named 'sma1-06.ddns.comp.nus.edu.sg' that provides the Oracle database reads, Buffer size (in MB) and concurrent user connections to a *Qmaster*, would be as shown in Figure 7.

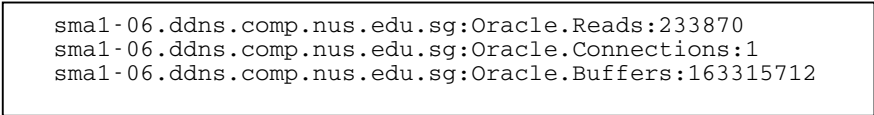


Fig. 7. Load/usage parameters captured by the Load Sensor

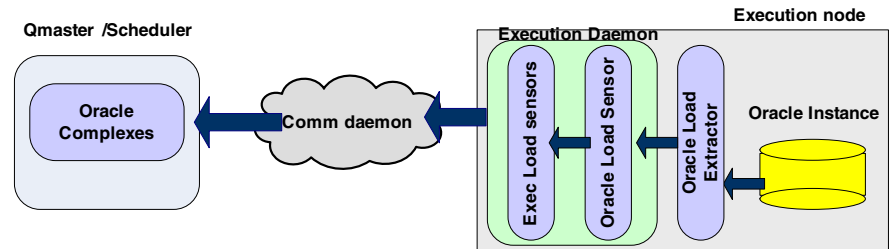


Fig. 8. Oracle Load /Usage details propagation

The difference in the load sensors and resource metadata is that resource metadata describes the resources available at the Oracle instance that is required for a job to be executed. In the other hand the Oracle load values are useful to dynamically select a low loaded node from the set of nodes that satisfy the resource requirements (tables, stored procedures and programs) of the job.

3.3 Resource Indexing in *Qmaster*

The current implementation of the *Qmaster* daemon maintains an execution host list. This is shown in Figure 9. For every host, a load list (EH_load_list) is maintained that holds all the values returned by the load sensor. This includes fields such as the load average (load_avg) and free swap (swap_free). Every 30 seconds, the *Qmaster* daemon will query the Execution daemon for these load values and does a refresh of its data. The load list will then be passed to the Scheduling daemon to build a Complex table that is used for matching with the attributes specified by the user when he submits a job.

The problem that we face here is that the existing load sensor returns single values for each attribute. For the Oracle procedures, tables and programs, they are multi-valued attributes, i.e. there can be tens of procedures and hundreds of tables. The existing data structures in the *Qmaster* and Scheduler daemons do not allow for an attribute to be in a list form. Therefore, we have changed the SGE code for both the *Qmaster* and Scheduler daemons to enable a list to be processed.

We have added 2 resource trees and 5 new load sensor fields into the SGE. These are the Oracle procedures, programs, tables, database reads, concurrent connections and buffer sizes. The reads, connections and buffers, are in the format of strings, thus they can be easily inserted into SGE from the *qmon* Graphical User Interface (GUI) as a user-defined complexes. For procedures, programs and tables, the existing data structure cannot be used as there are multiple procedures and tables for each host, thus the data cannot be stored as a string. It has to be augmented to allow list of procedures and tables to be stored.

The modified data structure now includes a schema list (HL_list). Both the Oracle procedures and tables share the same data structure. For each schema element, there is a procedure list (SCHEMA_list) that holds the procedure or table elements, each of which is identified by their procedure or table names (PT_name). The data structure for procedures is clearly illustrated in the Figure 9.

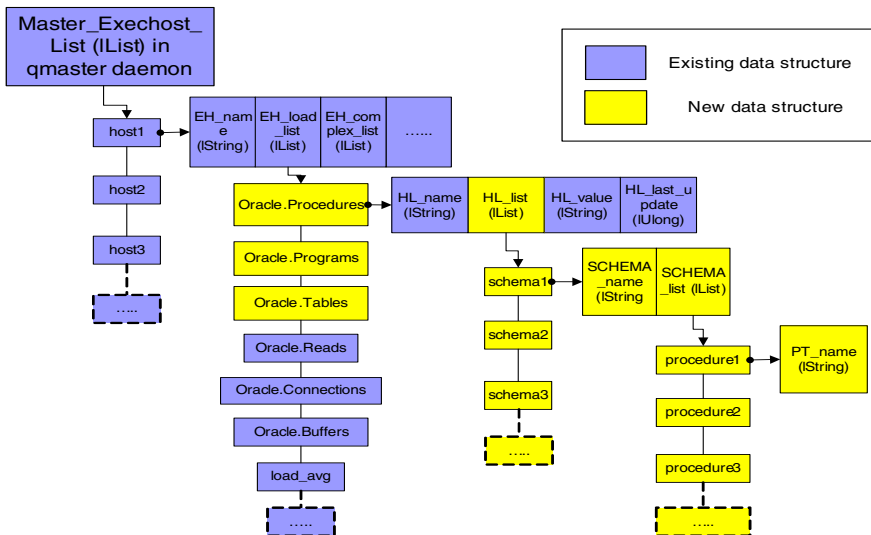


Fig. 9. New load list structure

Figure 10 shows part of the output of executing the command "qghost -F" that is obtained after the *Qmaster* daemon has finished indexing the load values. The load list will be used by the Scheduler daemon to fill up the Complex table during job attribute matching. The data structure of the Complex table is similar to that of the load list, so no further elaboration will be made.

```

hv:Oracle.Procedures=Schema=SYSTEM
| - SYSTEM.GET_LOADS
| - SYSTEM.INTERNAL_SURROGATE_SYSTEM
| - SYSTEM.LOADPROVIDER
| - ...
hv:Oracle.Programs=Schema=SYSTEM
| - SYSTEM.TESTSAJIPROGRAM
| - SYSTEM.TESTPROGRAM4
| - .....
hv:Oracle.Tables=Schema=SYSTEM
| - SYSTEM.AQ$_INTERNET_AGENTS
| - SYSTEM.AQ$_INTERNET_AGENT_PRIVS
| - SYSTEM.AQ$_QUEUES
| - ...
hl:Oracle.Reads=28101158.000000
hl:Oracle.Connections=2.000000
hl:Oracle Buffers=163315712.000000

```

Fig. 10. Output from “ghost -F” after Qmaster indexing

4 Conclusion and Future Work

In this paper, we incorporated Oracle resource discovery including tables, stored procedures and Oracle programs to the resource discovery and indexing mechanism of SGE. Further features were added to propagate Oracle specific load/usage information through the execution nodes to the SGE Scheduler. Thus SGE scheduler is able to consider the existence of Oracle resources and Oracle loads information retrieved from each execution host when deciding on where to dispatch Oracle jobs. SGE *Complex* [4] concept was extended from flat-single value resource to represent multi-valued hierarchical resource structures.

As the first stage, the implementation is based on SGE Enterprise and Cluster Grid environments. It could be extended to facilitate a Global grid environment with a possible API extension exposed for GRAM and GRIS services of Globus [3]. Furthermore, the Oracle Scheduler services we incorporated would not be able to be mirrored in a secondary SGE Scheduler/*QMaster*. Future work could be carried out for shadowing with minimal effort.

References

1. Oracle 10g data grid, <http://otn.oracle.com/tech/grid/index.html>
2. Sun Grid Engine, <http://gridengine.sunsource.net/>
3. Globus Toolkit, <http://www-unix.globus.org/toolkit/>
4. SGE Admin user Guide, <http://gridengine.sunsource.net/project/gridengine-download/SGE53AdminUserDoc.pdf?content-type=application/pdf>
5. SGE Load sensor architecture, <http://gridengine.sunsource.net/project/gridengine/howto/loadsensor.html>
6. SGE Qmaster service, <http://gridengine.sunsource.net/source/browse/gridengine/source/daemons/qmaster/qmaster.html?rev=1&content-type=text/html>
7. SGE Execution daemon Services, <http://gridengine.sunsource.net/unbranded-source/browse/%7Echeckout%7E/gridengine/source/daemons/execd/execd.html?content-type=text/html>

The Design and Implementation of a Locking Mechanism for a Distributed Computing Environment^{*}

Jaechun No¹, Hyo Kim², and Jang-sun Lee²

¹ Dept. of Computer Software
College of Electronics and Information Engineering
Sejong University, Seoul, Korea
`jano@sejong.ac.kr`

² System Software Laboratory, MacroImpact, Inc.
Gangnam-gu, Seoul, Korea
`{kshow,sunny}@macroimpact.com`

Abstract. The need for distributed file systems has been growing for decades to provide clients with efficient and scalable high-performance accesses to stored data. In this paper, we present a distributed locking mechanism that enables multiple nodes to simultaneously write their data to distinct data portions of a file, while providing the consistent view of client cached data, and conclude with an evaluation of the performance of our locking mechanism.

1 Introduction

Distributed file systems have been developed for decades to provide clients with efficient and scalable high-performance accesses to stored data. The clients are physically connected to one or more servers via a network like GigaEthernet or Fibre Channel [1–3, 5, 8, 9], and, on those clients, distributed file systems take responsibility for providing coordinated accesses to remotely stored data and for providing consistent views of client cached data. In such a distributed computing environment, one of major considerations affecting in achieving substantial I/O performance and scalability is to build an efficient locking mechanism.

A locking mechanism to support data consistency and cache coherency has a significant effect on generating high performance I/O. For example, large-scale scientific applications in physics, chemistry, biology, and other sciences generate huge amounts of data and utilize them for data analysis, visualization and so on. In order to achieve high-performance I/O, many such applications use parallel I/O methods where multiple client nodes simultaneously perform their I/O operations. MPI-IO is among those parallel I/O methods.

MPI-IO [6, 12] is specifically designed to enable the optimizations that are critical for high-performance parallel I/O. Examples of these optimizations include collective I/O, the ability to access noncontiguous data sets, and the ability

^{*} This research was supported by 2004 K*GRID program and MacroImpact Inc.

to pass hints to the implementation about access patterns, file-striping parameters, and so on. In order to achieve high I/O performance using MPI-IO on top of distributed file systems, the file system must provide the ability to lock a file per data section to have multiple concurrent writers to a file.

However, many of the locking mechanisms integrated with distributed file systems are based on a coarse-grained method [1–3] where only a single client at any given time is allowed to write its data to a file, while the other clients are waiting for the current node to finish its write operation even when the others would write to the different data portions of the same file. This drawback significantly degrades I/O performance in many scientific applications where supporting parallel write operations happens to be proved generating high I/O bandwidth [4, 6, 7, 10].

In this paper, we present a distributed locking mechanism based on multiple reader/single writer semantics for a data portion to be accessed. In this scheme, a single lock is used to synchronize concurrent accesses to a data portion of a file. However, several nodes can simultaneously run on the district data sections in order to support data concurrency. We conclude our paper by discussing performance evaluation of our locking mechanism.

2 Design Motivation

Our main objectives in developing a distributed locking mechanism were to provide high-performance parallel I/O, to minimize the communication latency occurred during the lock negotiation steps, and to utilize local lock services as much as possible.

- **High-performance I/O.** We designed the distributed locking protocol capable of allowing multiple concurrent writers to the same file to achieve high performance I/O. Also, the locking protocol provides data consistency between the data stored in the storage device and the data stored in the client-side cache.
- **Low communication latency.** We designed the locking protocol to reduce the network overhead taking place during the lock negotiation steps with Global Lock Manager (GLM). All the lock requests coming from the client nodes are evenly distributed on multiple GLMs. Moreover, in order to minimize the number of callback messages necessary to revoke and release a lock, we grouped all the client nodes into several node groups. If GLM finds the node group where the lock holder belongs to it then sends a lock revocation message to the node group.
- **Use of local lock service.** We designed the locking protocol to utilize local lock service to the maximum extents in order not to incur communication overhead with GLM and remote lock holders. By retaining the privileges on data sections even in the absence of active processes on a client, we eliminated the need to communicate with GLM repeatedly for the same data section, and thus can minimize the network latency.

3 Implementation Details

3.1 Overview of Distributed Locking Mechanism

When an application issues I/O requests using local file system interface, on top of VFS layer, each client should acquire an appropriate distributed lock from GLM in order to maintain data consistency for the cached data on clients and for remote, shared data on servers. The lock request is initiated by calling the lock interface, *snq_clm.lock*.

As mentioned in section 2, in order to reduce the communication latency occurring in the lock acquire step, we grouped the client nodes into several node groups. In the current implementation, an eight bit integer is used to denote node groups. When a client acquires an appropriate lock to perform I/O operation, the bit corresponding to the node group where the client belongs to is set to 1. Also, if a client requests a lock to GLM, GLM first locates the node group where the lock holder belongs to and then sends a callback message to the nodes of the node group. When the lock holder receives the callback message, it releases the requested lock and sends back an acknowledge to GLM to grant the lock to the requester.

Figure 1 represents a hierarchical overview of the locking construct with two client nodes and one GLM. The lock modes that we provide for are SHARED for multiple read processes and EXCLUSIVE for a single write process. The lock structure consists of three levels: metalock, datalock, and childlock. The metalocks, inode0 on node A and inode1 on node B in Figure 1, synchronize accesses to files and the value of a metalock is an inode number of the corresponding file. Below the metalock is a datalock responsible for coordinating access to a data portion. For example, on node A, metalock inode0 is split into two datalocks associated with the data sections 0-999 and 1000-1999 in bytes and, on node B, two datalocks below inode1 are associated with the data sections 0-2999 and 3000-5999 in bytes. In order to grant a datalock, the lock mode of the higher lock (metalock) must be SHARED, meaning that a file is shared between multiple clients.

The lowest level is a childlock that is of a split datalock. As mentioned in section 2, given that a datalock is granted, the datalock can be split further to maximize local lock services as long as the data section to be accessed by a requesting process does not exceed the data section of the datalock held. In other words, in Figure 1, the datalock for the data portion 0-999 is split into three childlocks that control accesses to the data portion 0-100, 100-199, and 800-899, respectively. The childlock is locally granted and therefore the requesting process need not communicate with GLM to obtain the childlock. However, the childlock is granted only when the lock mode of a childlock is compatible with that of the higher datalock. The datalock and childlock are found by comparing the starting file offset and data length being passed from the local file interface.

GLM contains the global lock information consisting of a list of locks that each GLM is responsible for serving. In Figure 1, GLM contains the metalocks, inode0 and inode1, and the datalocks of the data portions 0-999, 1000-1999, 0-

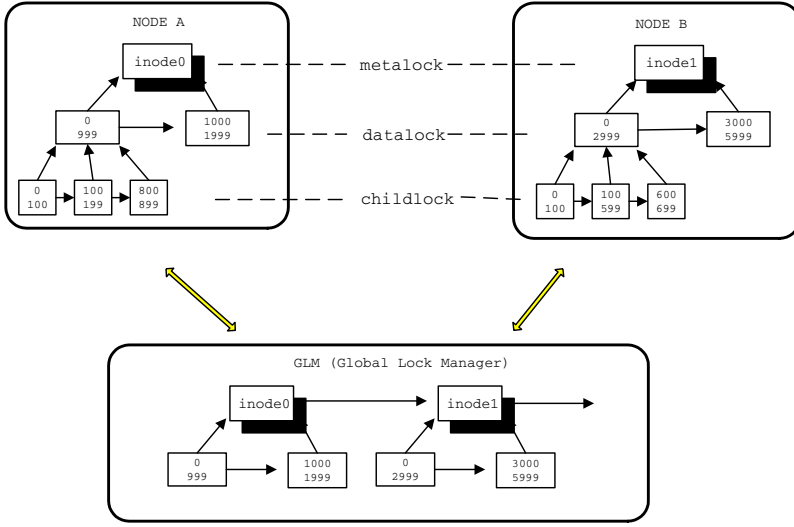


Fig. 1. A hierarchical overview of distributed locking mechanism

2999, and 3000-5999 held by node A and node B. GLM also contains the node group information indicating those groups where the lock holders belong to.

3.2 Function Calls of Distributed Locking Mechanism

Figure 2 represents the functions to be called to serve the lock request, lock release, and lock grant operations. The lock request operation is started by calling *snq_clm_lock* in the local file interface to read or write data. Once process finishes its I/O operation, *snq_clm_unlock* is called to wake up a sleeping process, if any, blocked while waiting for the lock to be released.

GLM receives the lock service request and then calls *glm_lock* or *glm_promote* to grant a lock or to upgrade lock mode. *Glm_lock* and *glm_promote* both call a callback invoke function, *glm2llm_callback*, to send an appropriate callback message to remote clients. *Glm2llm_callback* invokes *send_callback_msg* that sends a message to the node group where the lock holder belongs to. After invoking *send_callback_msg*, *glm2llm_callback* is blocked until it is woken up by *glm_unlock* or by *glm_demote*. *Glm_unlock* is a function to be called to update the global information of the lock that has been released on a remote lock holder and *glm_demote* is of the lock that has been downgraded on a remote lock holder.

On a client node, once a callback message is received, the lock interface calls *llm_callback* to release or to downgrade the lock requested. The lock release operation is performed by calling *llm2glm_unlock* and the lock downgrade operation is performed by calling *llm2glm_demote*. After completing its intended operation, each function sends back an acknowledge to GLM to grant the lock to the requesting node.

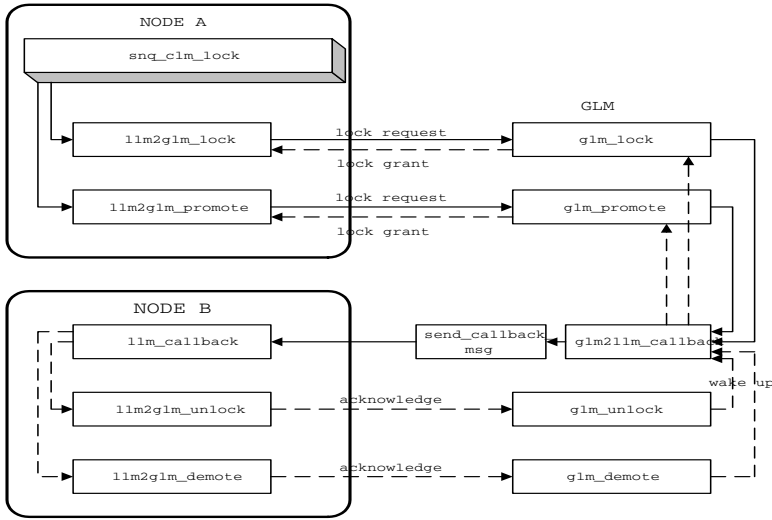


Fig. 2. Steps to acquire a distributed lock

4 Performance Evaluation

We measured the performance of the distributed locking mechanism on the machines that have Pentium3 866MHz CPU, 256 MB of RAM and 100Mbps of Fast Ethernet. The operating system installed on those machines was RedHat 9.0 with Linux kernel 2.4.20-8. The measurements include the times to take locks by performing lock revoke, downgrade, and upgrade operations, except for the times to invalidate client cached data and to write dirty data to disk.

Figures 3 and 4 represent the times to obtain the locks with the exclusive mode in write operations and with the shared mode in read operations, as the number of clients increases from 4 to 16. Also, in Figure 3, one machine was configured as a GLM and, in Figure 4, four machines were configured as GLMs. When four machines were configured as GLMs, each lock request is given to a GLM, according to round robin fashion. All clients read or wrote 1Mbyte of data to the distinct portions of the same file. In this case, the lock requested by each client is newly created on GLM and returned to the requesting client, causing no callback message to be sent to the remote lock holder to revoke the lock requested.

Figures 5 and 6 show the times to obtain the locks with the exclusive mode and with the shared mode, while moving each client's data section to access to the one of the next client at any given step, in order to observe the communication overhead occurred with the lock revocation on the remote lock holder.

Figures 5 and 6 both illustrate that the overhead of the lock revocation is significant with the exclusive mode because only a single client is allowed to write to a data section at any given time. With the shared mode, there is no need to contact the remote lock holder since a single lock can be shared between multiple

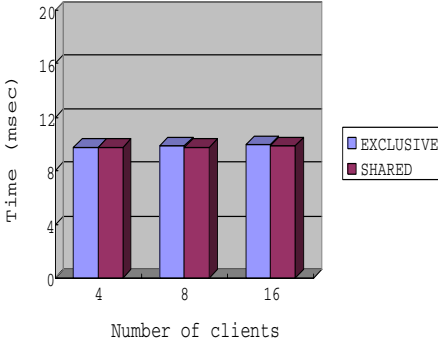


Fig. 3. Time overhead to acquire a distributed lock using one GLM. Each client read or wrote 1Mbytes of data to the distinct section of a file

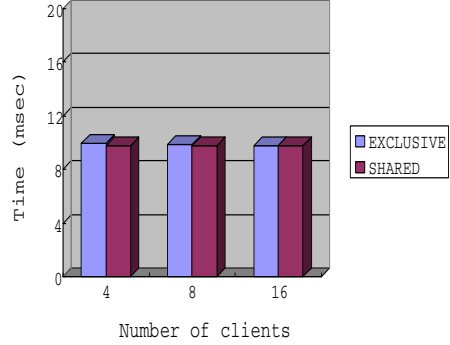


Fig. 4. Time overhead to acquire a distributed lock using four GLMs. Each client read or wrote 1Mbytes of data to the distinct section of a file

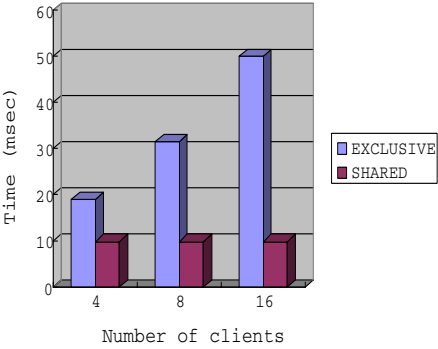


Fig. 5. Time to acquire a distributed lock using one GLM, while, at each step, changing a client's data section to access to the one of the next client

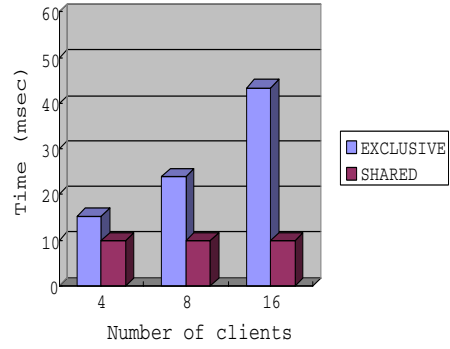


Fig. 6. Time to acquire a distributed lock using four GLMs, while, at each step, changing a client's data section to access to the one of the next client

nodes. With the shared lock mode, GLM just increases a counter denoting the number of shared lock holders before granting the lock.

Figure 7 shows the effect of childlocks exploiting locality in the lock requests. The lock locality ratio means how often childlocks are taken. Figure 7 shows that, with the exclusive lock mode, the more childlocks are generated, the smaller time is taken to serve a lock request due to the drop in time to negotiate with GLM and remote lock holders. With the shared lock mode, however, the time to take a lock flattens out at about 9 msec because the remote shared lock holders need not give up the requesting lock, allowing to have multiple lock holders with the shared mode. We believe that, however, more performance measurements must be conducted to verify the effect of lock locality.

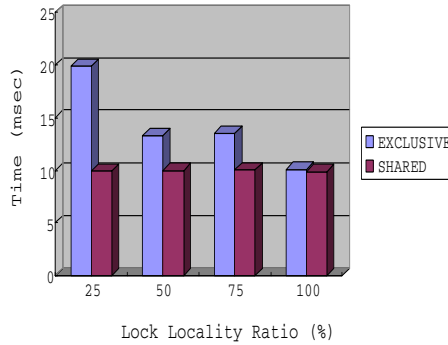


Fig. 7. Time to obtain a distributed lock as a function of lock locality ratio, using four clients with four GLMs

5 Conclusion

Concurrent accesses to the same file frequently occur in a distributed computing where allowing parallel write operations significantly improves I/O bandwidth. However, most distributed client-server file systems support a coarse-grained locking mechanism in which all the concurrent write operations to a file are serialized even when the data sections being written are different between writers. In this paper, we presented a distributed locking mechanism with which several nodes can simultaneously write to the distinct data portions of a file, while guaranteeing a consistent view of client cached data. The distributed locking mechanism has also been designed to exploit locality of lock requests to minimize communication overhead with GLM and remote lock holders. As a future work, we plan to integrate the locking scheme with a SAN-based cluster file system, called SANique, developed by MacroImpact company.

References

1. Murthy Devarakonda, Bill Kish, and Ajay Mohindra. Recovery in the Calypso file system. *ACM Transactions on Computer Systems*, 14(3):287–310, August 1996
2. Chandramohan A. Thekkath, Timothy Mann, and Edward K. Lee. Frangipani: A Scalable Distributed File System. In *Proceedings of the Symposium on Operating Systems Principles*, 1997, pages 224–237
3. Kenneth W. Preslan, Andrew P. Barry, Jonathan E. Brassow, Grant M. Erickson, Erling Nygaard, Christopher J. Sabol, Steven R. Soltis, David C. Teigland, and Matthew T. O’Keefe. A 64-bit Shared Disk File System for Linux. In *Proceedings of Sixteenth IEEE Mass Storage Systems Symposium Seventh NASA Goddard Conference on Mass Storage Systems & Technologies*, March 15-18, 1999
4. Jean-Pierre Prost, Richard Treumann, Richard Hedges, Bin Jia, Alice Koniges. MPI-IO/GPFS, an Optimized Implementation of MPI-IO on top of GPFS. In *Proceedings of Supercomputing*, November 2001

5. F. Schmuck and R. Haskin. GPFS: A Shared-Disk File System for Large Computing Clusters. In Proceedings of the First Conference on File and Storage Technologies (FAST), pages 231–244, Jan. 2002
6. Rajeev Thakur and William Gropp. Improving the Performance of Collective Operations in MPICH. In Proceedings of the 10th European PVM/MPI Users' Group Conference (Euro PVM/MPI 2003), September 2003
7. Rajeev Thakur, William Gropp, and Ewing Lusk. Optimizing Noncontiguous Accesses in MPI-IO. *Parallel Computing*, (28)1:83-105, January 2002
8. Philip H. Carns, Walter B. Ligon III, Robert B. Ross, and Rajeev Thakur. PVFS: A Parallel file system for Linux clusters. In Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, October 2000
9. P. J. Braam. The Lustre storage architecture. Technical Report available at - <http://www.lustre.org>, Lustre, 2002
10. Jaechun No, Rajeev Thakur, and Alok Choudhary. High-Performance Scientific Data Management System. *Journal of Parallel and Distributed Computing*, (64)4:434-447, April 2003
11. William Gropp, Ewing Lusk, and Rajeev Thakur. *Using MPI-2: Advanced Features of the Message-Passing Interface*. MIT Press, Cambridge, MA, 1999
12. Jean-Pierre Prost. MPI-IO/PIOFS. World-Wide Web page at <http://www.research.ibm.com/people/p/prost/sections/mpiio.html>, 1996
13. William Gropp and Ewing Lusk and Rajeev Thakur. *Using MPI-2: Advanced Features of the Message-Passing Interface*, MIT Press, 1999, Cambridge, MA

Replica Location Mechanism Based on DHT and the Small-World Theory

Xindong You, Guiran Chang, Wei Yang, Wandan Zeng, and Xueyao Chen

School of Information Science & Engineering, Northeastern University
110004 Shenyang, China
youxindong@sina.com.cn, chang@neu.edu.cn

Abstract. Data replication is a general mechanism to improve the performance and availability for distributed applications. To locate these replicas efficiently in a large-scale data grid system is a challenging task. In this paper we present a new replica location mechanism – Ridrop (Replica Information Drop), which is based on DHT and Small World Model. It employs the Gossip and Bloom Filter techniques to locate the replicas in the inner VOs domain, which is divided according to the feature of Small World. When the replicas of data are all beyond its own VO, we utilize DHT to locate or spread the replicas information at Home Nodes. Simulation experiment results show that Ridrop can achieve good performance in load balancing at Home Nodes.

1 Introduction

A data grid [4] is established for the data-intensive applications. For a large amount of data in the grid, it is necessary to create replicas in some grid nodes for the sake of the efficient data access, reliability and fault-tolerance of the system. However, how to locate these replicas is a main concern in large-scale data grid systems. The Small World model and DHT are helpful to solve this problem.

It is very common for the existence of short average path length. People can find it automatically. This is so-called “Small World” [1]. It widely exists not only in the society network but also in computer networks. File-sharing graphs in the D0 Collaboration [2] exhibit small-world characteristics: short average path lengths and large clustering coefficients. Although those file-sharing graphs are relatively small compared to the target of data grid, we expect similar usage patterns for the data grid, which is established for the same aim: sharing and analyzing the large amount of data.

On the other hand, DHT (Distributed Hash Tables) technology is widely used in P2P networks to locate resources [5]. It transforms a string of given length to a keyword through a certain hash function. Then the keyword is used to locate the resource or to store data in the nodes of the P2P networks. In this paper, we apply DHT technique to our system to locate or distribute the replicas information at home nodes.

Replica location technique has very important influence on data grids. The replica location mechanism presented by this paper integrates the techniques of DHT, Gossip [11], and Bloom Filter, based on the small-world model, with the properties of load balancing, extensibility and high efficiency.

2 Related Research

There are many location methods in P2P networks or in data grids. The data grid system—SRB[6] adopts the centralized meta-data servers to locate the data replicas. This approach inherits the disadvantage of the centralized system: single point of failure. The problem also occurs in the famous Globus toolkits, which provide replicas catalog service to locate replicas. Expansibility and reliability of these systems are limited. The Giggie [9] framework proposed by the Globus group and European Data [10] Grid group offers flexible replica location services, in which services' parameters can be configured to satisfy the clients' need to some extent. A decentralized, adaptive replica location mechanism presented in [7], in which Bloom Filter technique is employed to compress the whole replicas information in all RLNs (Replica Location Nodes), achieves good performance in querying, but imposes heavy storage load on RLNs. Another replica location mechanism named DSRL makes tradeoff between the query performance and storage consumption [8]. This approach is dynamic self-adaptive, scalable and reliable. It can join into or depart from home nodes adaptively. However, it can't deal with the situation of abrupt failure at home nodes, which leads to inaccurate location due to the lost of replica information at the failed home node. While the method presented in [3] assumingly partitions VOs based on Small World feature in P2P network. Then it employs different methods to locate data between inner VOs domain and inter VOs domain. Gossip [11] technique is utilized to spread the data information in the inner VOs domain. If the data is outside the domain, a bandwidth-consuming technique – unicast, multi-cast, or flooding, is employed. However, what topology protocols can induce the Small World in a certain cluster is not solved. They only discuss some possible directions for research.

In this paper, we try to solve some of the problems that the above-mentioned related researches have not settled. Such as:

- 1) *Improvement work done on the [3] to save the storage space of the system.*
- 2) *Form the principle to induce the Small World phenomenon in a certain VO.*
- 3) *Employing DHT technique to even the load on the home nodes and realize the genuine decentralization of the system.*

3 A Replica Location Mechanism – Ridrop

To have a uniform interface to data accessing and data management, we consider all the data in a data grid to be data elements (DE). Each data element has a global unique logic data name (LDN) and a physical data name (PDN). The replicas of the same data element share the same LDN, while the PDNs of the replicas are different. To locate the data replica(s) means to map LDN of the data element to its PDN (s).

3.1 The Model of Replica Location Mechanism

The replica location mechanism model we constructed includes two steps. Firstly we divide the VOs based on the feature of Small World, and then collect all the VOs.

To shape Small World in the VOs, WSDL is used to describe the property of VOs and the data in the VOs called VODLs (VO description language) [12], which are checked by any node that will be joined into the grid system. Under the consumption, the requirement of the data elements is to be in high similarity in a certain VO. The description of the property of the VOs in WSDL (VODL) is as follows:

```
<? xml version="1.0" encoding="UTF-8"?>
<vodl:VirtualOrganizationDescription name="StorageVO"
element="vodl:String" mutability="mutable">
  <wsdl:documentation>
    The Description of the Virtual Organization
  </wsdl:documentation>
</vodl:VirtualOrganizationDescription>
<VODataSet>
  <VOData name="vodl:DataType">
    <wsdl:documentation>
      The Description of the dataType
    </wsdl:documentation>
    <vodl:RequiringRate>0.252</vodl:RequiringRate>
  </VOData>
  <VOData name="vodl:DataType">
    <wsdl:documentation>
      The Description of the dataType
    </wsdl:documentation>
    <vodl:SupplyingRate>0.280</vodl:SupplyingRate>
  </VOData>
  <VOData name="vodl:basiclocation">
    <vodl:String>"http://172.16.2.187/Ridrop"</vodl:String>
  </VOData>
  <VOData name="vodl:nodesNum">
    <vodl:int>182343</vodl:int>
  </VOData>
</VODataSet>
<VODataSet>
  ...
</VODataSet>
```

In the VODL RequiringRate means the probability of the VO requiring the data of the node, while the SupplyingRate is the data of the VO can supply to the joining node. When a node wants to join into the grid system, it checks the VOs' VODLs. By comparing the RequiringRate and the SupplyingRate among the VODLs, the node joins the VO with either the highest RequiringRate or the highest SupplyingRate or relatively high of both. As you can see, with the VODLs the principle is formed to induce the Small World phenomenon in a certain VO. Connecting all the VOs, our system model is built up as shown in figure1.

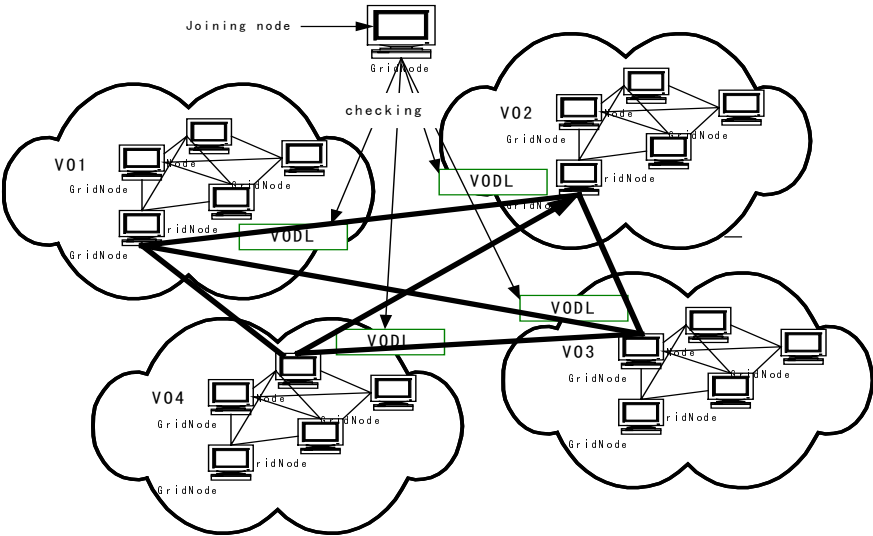


Fig. 1. The System Model of the Replica Location Mechanism

3.2 Replica Location Method in the Inner Domain

Before talk about the replica location method in the inner Domain, a definition is given firstly.

Def 1 stable VOServers: a set of grid nodes in a certain VO with the properties of stability, high capacity, powerful process ability and broad bandwidth. Stable VOServers are selected among grid nodes in the certain VO. Additional VOServer will be elected among remain grid nodes whenever one of the VOServers fails.

In [3] Gossip technique is employed to disseminate the data information to neighboring node in the inner VO domain. After a period of time, every node gets the data information of all the other nodes in its VO. And then Bloom Filter technique is utilized to compress the whole data information in each node, which imposes heavy storage load on the system, although the compression technology is employed. Furthermore, storing the whole data information at each node makes it a challenging problem to update the data information. Some improvement has been made in this paper on the basic idea of [3]. After a node picks up the data's replicas information of all the other nodes in its VO, the whole data's replicas information is not stored at

each node, but at some stable VOServers in the inner VOs. Whenever a grid node in the VO desires to inquire the replicas, the request is forwarded to the VOServers. Locating the physical replicas is the task of one of the VOServers. They also will be informed whenever the replicas information is changed in any node. Consequently, it is much easier to update the whole replicas information as compared to [3]. Meanwhile, to save storage space further, Bloom Filter technique is also employed to compress the whole replicas information in the VOServers. As a result, the method presented here can save more storage space than [3] with no reliability lost.

3.3 Inter domain Replica Location Method

When the replicas requested by a node are not in its VO, the inter domain replica location mechanism will be applied. Before describing the mechanism, a definition is given below.

Def 2: home node: A node storing the replicas information of a data element is called the home node of the data element. A home node might store replicas information of many data elements. It takes the responsibility for locating physical replicas of the data element whose replicas information is out of its inner VO.

For each data element in a data grid system, the MD5 function is applied to its LDN to produce an identifier with 128 bits. We call the identifier the Global Data Identifier (GDI). It can be described as follows:

$$\forall O, \exists \text{GDI}(O) = \text{MD5}(\text{LDN}(O)) \quad (1)$$

In (1), O represents a Data Element and GDI may be thought as the unique address due to the property of MD5 function. On the other hand, each home node is assigned a unique address named GHA (Global Home Address), which might be the grid node's address being used in a data grid system.

Now, we can query the replicas information through the home nodes by the DHT technique. When the replicas of a data element are all beyond its VO, the request for the replicas information is forwarded to the home node that is mapped by a certain hash function. That is to say: Forward (InfoRequest) to

$$\text{GHA}(O) = \text{Hash}(\text{GDI}(O)) \quad (2)$$

Meanwhile, the changed replicas information of any data element whose replicas are beyond its VO is also forwarded to the according home node with GHA. In other words: Forward (changedInfo) to (2).

So the replicas information at the home nodes is assured to be consistent and accurate. But when it comes to a problem with large-scale data elements that request to locate the replicas information at home nodes, load balancing on home nodes is of very importance. The simulation experiment results in next section show that employing DHT technique achieves good performance in load balancing on the home nodes. To avoid a sudden failure on home nodes, additional backups of replicas information are also stored on the logic neighboring home nodes, which makes our mechanism reliable. The locating process is sketched in Figure 2.

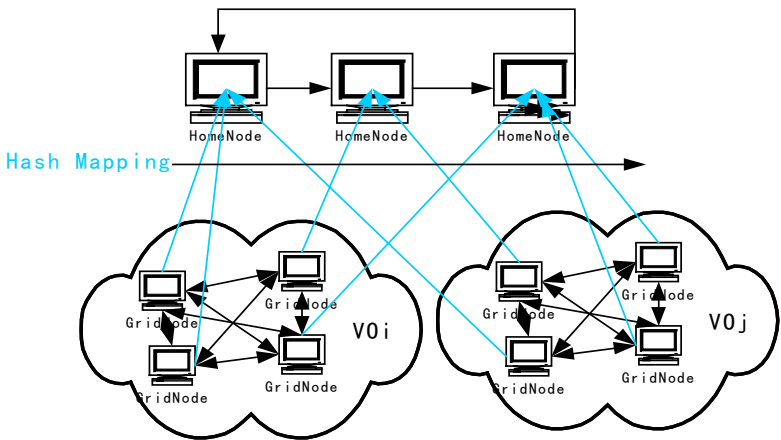


Fig. 2. Inter Domain Replica Location Mechanism

4 Simulation Experiment Results

In the simulation experiment, we assume there are 5000, 50000, 50000 data elements request to locate replicas information at 50 home nodes respectively. Loads on these home nodes are evaluated accordingly. The simulation experiment is carried on a Java program in which a link list with 50 objects represents the 50 home nodes. The hash function implemented on the LDN produces an identifier. The identifier will be mapped to one of the home nodes. Load on a home node is expressed as access times of a data element to mapped home node.

The DSRL replica location mechanism proposed in [8] is proved in theory that the heaviest load on home nodes is not much more than $2I/N$ (I represents the number of the data elements needed to locate replicas information, while N represents the number of home nodes). Meanwhile, the heaviest load is not twice more than the lightest load on the home node. In the following figures RI denotes Replicas Information.

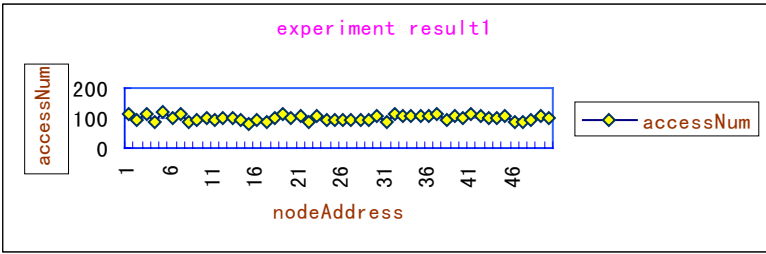


Fig. 3. Loads on the home nodes when there are 5000 data elements request the RI

From Fig3, the home node with the heaviest load needs to process locating replicas information of 117 data elements while the node with the lightest load 81 data elements. From the results we know:

First :heaviest load =117 << $2I/N=2 \times 5000/50=200$.

Second : heaviest load=117< $2 \times \text{lightest load}=2 \times 81=162$.

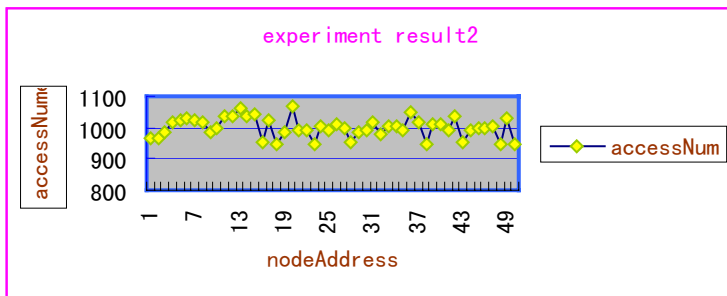


Fig. 4. Loads on the home nodes when there are 50000 data elements request the RI

The simulation results in Fig 4 show that:

First : heaviest load =1079 << $2I/N=2 \times 50000/50=2000$.

Second : heaviest load=1079<< $2 \times \text{lightest load}=2 \times 944=1688$

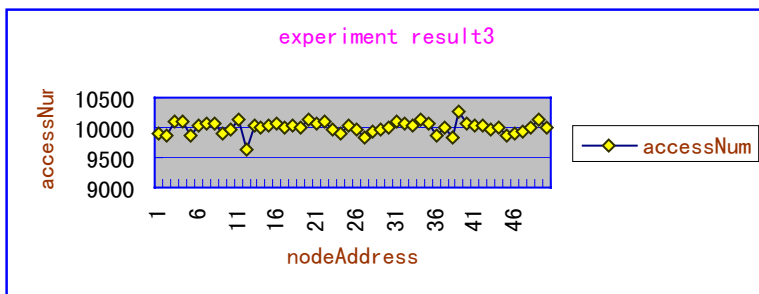


Fig. 5. Loads on the home nodes when there are 500000 data elements request the RI

The simulation results in Fig 5 show that:

First : heaviest load =10278 << $2I/N=2 \times 500000/50=20000$.

Second : heaviest load=10278<< $2 \times \text{lightest load}=2 \times 9625=18200$.

From the above three simulation experiment results we conclude that the loads on the home nodes are relatively balanced on the whole regardless of the data scale, which is important to the growing data grid system, from which we can deduce that the scale of the each VO will make little influence on the performance of the whole system. And by comparison to [8], the Ridrop mechanism performs better in load balancing in some degree.

5 Conclusions and Future Work

This paper has presented a new replica location mechanism, which is based on the DHT technique and the Small World model. It bears the properties of load balancing,

reliability, genuine decentralization and extensibility. Most replicas can be located in the inner VO domain when the VOs are partitioned according to the Small World model. It reduces the frequency to locate replicas beyond the domains and improves the performance of the whole system. Furthermore the decentralization is achieved through the employing of the DHT technique.

One of the future tasks is to design more effective hash functions for each VO according to its characteristic. The other is to investigate the Small World in more detail in each VO to design more reasonable VODLs by which the joining node can find its VO more efficiently.

Acknowledgements

This work was support the National research Foundation for the Doctoral Program of Higher Education of China under Grant No.20030145017.

References

1. S. Milgram. The Small World problem. *PSYCHOLOGY Today*. 1967, 67(1); 60-67.
2. The Experiment, <http://www-d0.fnal.gov>.
3. A. Iamnitchi, M. Ripeanu, I. Foster. Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations. In *Proceedings of MultiMedia Computing and NetWorking 2002 (MMVN'02)*.
4. W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23:187-200, 2001.
5. YANG Wei. 2004. A DHT-Based Hierarchical Peer Group Lookup Method in Peer-to-Peer Systems. THESIS FOR MASTER'S DEGREE OF NORTHEASTERN UNIVERSITY.
6. C. Baru, R. Moore, A. Rajaseker et al. The SDSC storage resource broker. In: *Proc of a) CASCON'98*. Toronto, Canada: IEEE Computer Society Press, 1998.
7. M. Ripeanu, I. Foster. A Decentralized, Adaptive Replica Location Mechanism. In: *Proc of HPDC-11*. Edinburgh, Scotland: IEEE Computer Society Press, 2002.
8. LI Dong-sheng, LI Chun-Jiang, XIAO Nonf, WANG Yi-Jie, and LI Xi-Chang. Dynamic Self-Adaptive Replica Location Method in Data Grids. *JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT*. Vol .40, No.12, Dec.2003.
9. A. Chervenak, E. Deelman, I. Foster et al. Giggles: A Framework for Constructing Scalable Replica Location Service. In: *Proc of SuperComputer 2002 (SC2002)*. Baltimore, USA: IEEE Computer Society Press, 2002.
10. European datagrid project. www.eu-datagrid.org.
11. A. Kermarrec, L. Massoulié, and A. Ganesh. Reliable probabilistic communication in large-scale information dissemination systems. Technical Report MSR-TR-2000-105, Microsoft Research Cambridge, Oct.2000.
12. HANG Er-Fan, DU Zhi-Hui. Efficient Grid Service Location Mechanism Based on Virtual Organization and the Small-World Theory. *JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT*. Vol .40, No.12, Dec.2003.

A New Chameleon Multi-signature Based on Bilinear Pairing

Chunbo Ma and Dake He

Lab. of Information Security and National Computing Grid
Southwest Jiaotong University
Chengdu, Sichuan, P.R. China, 610031
cb_ma@263.net, hedk@home.swjtu.edu.cn

Abstract. With the combination of vector space secret sharing and Chameleon function, a new Chameleon multi-signature based on bilinear pairing is presented in this paper. The scheme has following properties: only the appointed receiver can verify the signature; the appointed receiver can't disclose contents of the signed information to any third party without the signer's consent; undeniable; when dispute occur, the signer can prove the signature to be forged without exposure the origin signature; it can protect the signature from allied cheating which comes from in or out the group. With the security analysis, we can conclude that the signature is secure.

1 Introduction

Krawczyk and Rabin first proposed the Chameleon signature [1] in 2000. The signature differ greatly from other ones is the Hash function. Chameleon Hash function is a type of trapdoor one-way function. The signature constructed by this type of Hash function is an appointed one, viz. only the appointed receivers can verifier the signature. After the Chameleon signature was proposed, another scholar given a scheme based on bilinear pairing [2].

D. Boneh and M. Franklin, bring the bilinear pairing into encryption, proposed a short signature scheme [6] based on bilinear pairing in 2001. The signature has the properties of short, security and high efficiency. Since then on, the bilinear pairing arrest the scholars attention and lots of researches based on it. X. Chen, F. Zhang and K. Kim, bring the bilinear pairing into signature schemes based on ID, proposed a high efficient group signature. The security of this signature [7] is equal to solve the discrete logarithm on ECC. The secret sharing and the bilinear pairing were combined in [8] and a new threshold blind signature was proposed.

With the combination of vector space secret sharing and Chameleon signature, a new Chameleon multi-signature based on bilinear pairing is presented in this paper. The scheme has following properties: only the appointed receiver can verify the validity of the signature; the appointed receiver can't convince the third party of the validity of the signature; undeniable; when dispute occur, the signer can prove the signature to be forged without exposure the origin signature; it can protect the signature from allied cheating which comes from in or out the group.

2 Bilinear Pairings

Let G_1 be a cyclic additive group generated by P , whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q . Assume that the discrete logarithm in both G_1 and G_2 is hard. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ and satisfies the following properties:

1. Bilinear: $e(aP, bP') = e(P, P')^{ab}$. For all the $P, P' \in G_1$ and $a, b \in \mathbb{Z}_q$, the equation holds
2. Non-degenerate: There exists $P' \in G_1$, if $e(P, P') = 1$, then $P = O$
3. Computable: For $P, P' \in G_1$, there is a efficient algorithm to compute $e(P, P')$

When the problem of CDHP (Computational Diffie-Hellman) is hard but of DDHP (Decision Diffie-Hellman Problem) is easy, the group G called GDH (Gap Diffie-Hellman Group). This type of group can be constructed in the field of Hyper Elliptic or Super Singular Elliptic Curve [3]. The bilinear pairing can be derived from the Weil or Tate pairings.

The symbol mentioned above will be used in the following text.

3 Vector Space Secret Sharing

The vector space secret sharing scheme mentioned in [5] is briefly reviewed as follows.

Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of n participants, Γ be the set of the subset of P . If any subset in Γ can make out the secret k , we call Γ accessed structure and call the subset authorized subset.

Vector space secret sharing is a type of perfect scheme for accessed structure. Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of n participants, Γ be accessed structure and $D \notin P$ be the trusted center. Let $K = GF(q)$, where q is a large prime number, K^r denote the vector space consist of all the r elements on K . If there is a function $\varphi : P \cup \{D\} \rightarrow K^r$ satisfied the following property, we call Γ a vector space accessed structure:

$$\varphi(D) = (1, 0, \dots, 0) \in \{\varphi(p_i) = (a_{1,i}, a_{2,i}, \dots, a_{n,i}) : p_i \in A\} \Leftrightarrow A \subset \Gamma \quad (1)$$

In other words, vector $\varphi(D)$ can be expressed as vector's linear combination in set $\{\varphi(p_i) : p_i \in A\}$, if and only if A is an authorized subset.

If Γ is a vector space access structure, for all the $p \in P$, $S_p \in K$ (S_p denote all the possible sets of sub-secrets of participant p may be get), a perfect secret sharing scheme can be established. Let $k \in K$, distributor randomly selects $v_2, v_3, \dots, v_r \in K$. Let $V = (v_1, v_2, \dots, v_r)$, $v_1 = k$, then $(V, \varphi(D)) = k$. The sub-secret distributed to participant i is $w_i = (V, \varphi(p_i))$, viz. $w_i = \sum_j v_j a_{ji}$.

Function φ is public. The participants in authorized subset can work out the secret k by computing the linear combination of the sub-secret they own. In fact, assume that $A = \{p_1, p_2, \dots, p_l\}$ is an authorized subset, then $\varphi(D) = c_1 \varphi(p_1) + c_2 \varphi(p_2) + \dots + c_l \varphi(p_l)$, where $c_i \in K$. The participants in A can work out the secret k as $k = c_1 w_1 + c_2 w_2 + \dots + c_l w_l$. The scheme construct in this way called vector space secret sharing scheme.

4 Vector Space Secret Sharing Chameleon Multi-signature

4.1 Initialize

Let the information m be the message to be signed, G_1 be the GDH of order q . Where q is a large prime number. The bilinear pairing can be defined as $e: G_1 \times G_1 \rightarrow G_2$.

TTP (Trusty Third Party) randomly selects secret $k \in Z_q$ and $v_2, v_3, \dots, v_l \in K$. Let $V = (v_1, v_2, \dots, v_l)$, $v_1 = k$. With the statement above, assume that $A = \{p_1, p_2, \dots, p_l\}$ is an authorized subset, then $k = c_1 w_1 + c_2 w_2 + \dots + c_l w_l$. $c_i \in K$ can be computed by every body. k and vector V should keep secret. TTP publicizes $R = kP$ and distributes the sub-secret to every participant and publicizes $R_i = w_i P$. The public key and the private key of TTP are vP and v respectively, where $v \in Z_q$ is random selected by TTP. The private key of receiver B is $vH_0(ID_B)$ and public key is $H_0(ID_B)$, where the ID_B is the identity of B.

Suppose that there are two one-way functions:

$$H_0: \{0,1\}^* \rightarrow G_1 \quad H_1: \{0,1\}^* \rightarrow Z_q^*$$

4.2 Individual Signature Generation

Suppose that m will be signed by participant P_i . The P_i generates the one-way Chameleon Hash function and the value s_i :

$$t_i = e(w_i P, P) e(H_1(m) H_0(ID_B), vP) \quad (2)$$

$$s_i = H_1(t_i) w_i H_0(m) \quad (3)$$

The signature of the participant for the information m is (m, s_i, R_i) . The signature will be sent to B through the TTP.

4.3 Verification of Individual Signature

The receiver B first computes t_i by the signature and verifies it with the following equation:

$$e(s_i, P) \stackrel{?}{=} e(H_1(t_i) H_0(m), R_i) \quad (4)$$

Only the designated receiver B can correctly compute t_i , so it is only B that can get $e(H_1(t_i) H_0(m))$ and verifies the above equation.

Theorem 1. *The signature is valid, if the equation (4) holds*

Proof.

$$\begin{aligned} e(s_i, P) &= e(H_1(t_i) v H_0(m), P) \\ &= e(H_1(t_i) H_0(m), R_i) \end{aligned}$$

4.4 Multi-signature Generation

If all the individual signatures are valid, the TTP do computation as follows:

$$T = \prod_{i=1}^l t_i^{c_i} = e\left(\sum_{i=1}^l c_i w_i P, P\right) e\left(\sum_{i=1}^l c_i H(m) H_0(ID_B), vP\right) \quad (5)$$

$$S = H_1(T) \sum_{i=1}^l c_i w_i H_0(m) \quad (6)$$

TTP sent the signature (m, R, S) to receiver B.

4.5 Verification of Multi-signature

The receiver B can verify the signature (m, R, S) with the following equation:

$$e(S, P) \stackrel{?}{=} e(H_1(T) H_0(m), R) \quad (7)$$

Theorem 2. *The signature is valid if the equation (7) holds.*

$$\begin{aligned}
 \text{Proof.} \quad & e(S, P) \\
 &= e(H_1(T)kH_0(m), P) \\
 &= e(H_1(T)H_0(m), R)
 \end{aligned}$$

5 Security Analysis

Below we analysis the six main ingredients about the security of the signature.

1. The signature fits for the non-continuous messages, for example, in the electronic auction. The system should changes all the parameters k, w_i for the next signature to keep R_i and R stochastic.
2. Only the receiver B can verify the validity of the signature in this scheme. In the equations (4) and (7), the signature can be verify only under the condition of computing the t_i correctly. Obviously, only the appointed receiver B can work it out.
3. When the disputes occur, suppose that the appointed receiver B forges the multi-signature (m', R', S') and pass the verification. In order to reveal the forge, the signer can do the following computation:

$$\begin{aligned}
 e(kP, P)e\left(\sum_{i=1}^l c_i H(m)H_0(ID_B), vP\right) &= \\
 e(k'P, P)e\left(\sum_{i=1}^l c_i H(m')H_0(ID_B), vP\right) & \quad (8)
 \end{aligned}$$

$$kP - k'P = \sum_{i=1}^l (c_i H(m') - c_i H(m))vH_0(ID) \quad (9)$$

$$vH_0(ID) = (R - R') / \sum_{i=1}^l (c_i H(m') - c_i H(m)) \quad (10)$$

If signer can get the correct private key of B, viz. $vH_0(ID)$, then the signer can give another signature (m'', R'', S'') different from (m', R', S') :

$$R'' = vH_0(ID) \sum_{i=1}^l c_i (H(m') - H(m'')) + R' \quad (11)$$

The signer can submit the TTP a validate signature different from the original one. This can make the signer to reveal the cheating and at the same time protect the original signature.

4. The signature has the property of undeniable, viz. the signer can't deny the signature come from him. With the statement in c), under the condition of unknown the private key $vH_0(ID)$ of receiver B, the signer can't generate a signature different from the original one. The difficulty of deriving v from vP equals to solve the problem of discrete logarithm.
5. With the properties of the secret sharing, the ally of any members under the threshold can't get the secret k . The difficulty of deriving k from kP equals to solve the problem of discrete logarithm.

6 Conclusion

With the combination of vector space secret sharing and Chameleon signature, a new Chameleon multi-signature based on bilinear pairing is presented. The scheme has follows properties: only the appointed receiver can verify the validate of the signature; when dispute occur, the signer can prove the signature been forged without exposure the origin signature; it can protect the signature from allied cheating came from in or out the group. With the security analysis, we can see that the signature is secure.

References

1. H. Krawczyk and T. Rabin. Chameleon signatures. In Proceedings of NDSS2000, 143-154, 2000
2. Fangguo Zhang and Reihaneh Safavi-Naini and Willy Susilo. ID-Based Chameleon Hashes from Bilinear Pairings. <http://eprint.iacr.org/2003/208>
3. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Advances in Cryptology – Asiacrypt'2001*, Lecture Notes in Computer Science **2248**, Springer-Verlag (2002), pp. 514–532
4. C. X. Xu, X. F. Tong, Z. G. Xiao. A Vector Space Secret Sharing Against Cheating. Journal of Xi Dian University, 2002, 29(4)pp.257–260
5. Benoh Libert and Jean-Jacques Quisquater. Identity Based Undeniable Signatures. <http://eprint.iacr.org/2003/206>
6. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Advances in Cryptology – Asiacrypt'2001*, Lecture Notes in Computer Science **2248**, Springer-Verlag (2002), pp. 514–532
7. X. Chen, F. Zhang, K. Kim. A New ID-based Group Signature Scheme from Bilinear Pairings. Cryptology ePrint Archive
8. D. L. Vo, F. Zhang, K. Kim. A New Threshold Blind Signature Scheme from Pairings. *2003 Symposium on Cryptography and Information Security (SCIS2003)*, Itaya, Japan, Jan. 26–29, 2003, vol. 1/2, pp. 233–238

A Topology-Adapted Network Defense Model Based on Mobile Agent

Yichuan Jiang, Yiping Zhong, and Shiyong Zhang

Department of Computing & Information Technology, Fudan University
Shanghai 200433, China
jiangyichuan@yahoo.com.cn

Abstract. Since the topology of now network system is always dynamic, the paper provides a network active defense model that is adaptive for dynamic topology based on the mobile agent technology. The model includes three parts: network topology discovery, adaptive agents modulation mechanism and active defense. The model provided by the paper contains two kinds of agents: topology discovery agent and defense one. The model uses mobile network topology discovery agents to actively probe the current network topology and encodes it; then the adaptive modulation part of the model implements the distribution and migration of the defense agents according to the current topology; at last the defense agents then make active defense for the network.

1 Introduction

At present, the many network defense technologies often can only be used on fixed topology in practice. When the network topology is transformed, we have to make many modulations for the existing network defense system, such as monitoring place, mechanism, strategy, etc. Therefore, the existing network defense system often can't exert its function well when the network topology is transformed, and a network defense system aiming at a kind of topology can't fit for other different topologies.

There are some projects engaged in the adaptation of network defense technology, such as JAM [1], GASSATA [2]; AAFID [3]; JPA [4]; MAIDS [5].

However, these related works don't make really systemic research on the adaptation for dynamic network topology. To solve such problem, we presented an original intrusion detection model that can distribute its agent resource according to the network topology in [6]. On the base of [6], now we present a topology-adapted network defense model, which is based on mobile agent technology.

The rest of the paper is organized as follows: Section 2 provides the basic model; Section 3 addresses the agent adaptive modulation mechanism; the last section makes an experiment and conclusion.

2 Basic Model Architecture

The model provided by the paper is based on mobile agent technology. There are two kinds of agents in the system: topology discovery agent and defense agent. And the

defense agents include intrusion sensor agent, intrusion detection agent, tracing agent and recovery agent. Aiming at the dynamic topology, we should firstly discover the current network topology timely and correctly based on topology discovery agent; after discovering the topology, the system then make adaptive modulation to the defense agents; then the modulated defense agents make active defense for the network.

3 Agent Adaptive Modulation Mechanism

We apply genetic algorithms and ant algorithms into the modulation mechanism, and propose a module which can adaptively modulate defense agents according to the topology.

3.1 Apply Genetic Algorithms to Implement Initial Agents Distribution

Given that there are n nodes, then the complete graph constructed by these nodes has $n(n-1)/2$ edges. Now we number the edges and nodes of the graph. According to the number of edges, the length of code is $n(n-1)/2$. Referring to the factual topology, the bit of the code can be 0 which signifies that the edge of the graph doesn't correspond to the one of the factual topology, or be 1 which signifies that the edge of the graph corresponds to the one of the factual topology. For example, a complete graph with 4 nodes is shown in Fig 1, the number of each edge is: 1(1,2), 2(1,3), 3(1,4), 4(2,3), 5(2,4), 6(3,4). And the factual network is shown in Fig 2, which is composed of edges {1,4}, therefore we can encode the factual topology as: {100100}.

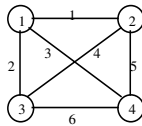


Fig. 1. A Complete Graph with 4 Nodes

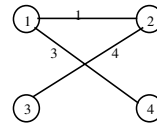


Fig. 2. The Factual Topology

In the model, while the network is intruded, firstly we should distribute the agents. Therefore, we should realize optimal distribution of defense agents according to current network topology, intrusion information and former agents distribution. Now we use genetic algorithms to implement such task.

(1) Encoding

We adopt subsection code. The chromosome is parted into 3 sections of gene. The first section denotes network topology, the second one denotes agent distribution state, the third one denotes intrusion information. Therefore, the chromosome is shown as follows:

$$a_1, a_2, \dots, a_i, \dots, a_{n(n-1)/2} \mid I_{11}I_{12}I_{13}I_{14} \ I_{21}I_{22}I_{23}I_{24} \ \dots \ I_{i1}I_{i2}I_{i3}I_{i4} \ \dots \ I_{n1}I_{n2}I_{n3}I_{n4} \mid b_1, b_2, \dots, b_i, \dots, b_n \quad (n \text{ is the number of nodes in the network})$$

Among those the first section is addressed in Section 3.2. The length of this section is $n(n-1)/2$.

The second section ' $I_{i1}I_{i2}I_{i3}I_{i4}$ ' shows that on the node i there are No. 1 agents (intrusion sensor agent) with the amount of I_{i1} , No. 2 agents (intrusion detection agent) with the amount of I_{i2} , No. 3 agents (tracing agent) with the amount of I_{i3} , No. 4 agents (disaster recovery agent) with the amount of I_{i4} . The length of this section is $4n$.

The third section is binary that shows which node is suffered from abnormal activity or not. If a node is suffered from abnormal activity then the corresponding bit in the code is 1, or the bit is 0. $i \in [1, n]$. The length of this section is n .

Therefore, the total length of the chromosome is:

$$n(n-1)/2 + 4n + n = \frac{n^2}{2} + \frac{9}{2}n \quad (1)$$

(2) Design the Fitness Function

When agent i moves from now location to the destination location, the migration cost includes resource cost and time one.

We design the migration cost function of agent i as following:

$$Cost_i = h(\sigma_1 C_t + \sigma_2 C_r) \quad (2)$$

Among those C_t is the time cost when agent moves from a node to it's adjacent node, C_r is the system resource cost when agent moves from a node to it's adjacent node. h is the hops when agent moves from the now location to the destination location. σ_1 and σ_2 are the weight of C_t and C_r . How to compute h and the weight is out of this paper, if the reader is anxious he can refer to [9]. The adaptive act of agent i should make the cost function lest.

In genetic algorithms, we often use a non-negative real number to reflect the fitness ability of individual. In order to adjust to the character of genetic algorithms and combining our code and the above cost function, we can define fitness function as:

$$F^t = F_0^t - \sum_{i=1}^N Cost_i \quad (3)$$

Among those N is the amount of mobile agents. F_0^t is a positive constant, which can changes along with the problem size and is used to ensure individual fitness F^t always non-negative.

(3) Production of Initial Population

In term of the encoding method, we produce initial population with the individual length of $\frac{n^2}{2} + \frac{9}{2}n$. Among those the first $n(n-1)/2$ bits of gene is binary; the middle $4n$ bits of gene is natural number or 0; the last n bits of gene is binary.

(4) Genetic Operator

Selection: According the Equation (3), we can compute the fitness of individual and select the individual with high fitness. Crossover: we use crossover operation to produce new individual; Mutation: we also use mutation to produce more robust individual.

3.2 Apply Ant Algorithms to Implement Agent Migration

3.2.1 Residence Factor of Agent

Firstly we design a array $A[k][i]$ to denote that the number of successful defense while agent k locates at node i . After a network defense system is initially installed in a network, $A[k][i]$ is zero. Once the network defense system make defense successfully, we add $A[k][i]$ with 1, or else we subtract $A[k][i]$ with 1. However, $A[k][i]$ can't be less than zero.

Def 1. Residence factor of agent k at node i is defined as followings:

$$res_k(i) = \ln(A[k][i] + 1) \quad (4)$$

From Equation (4), we can see that when $A[k][i]$ is zero, then the residence factor of agent k at node i is zero.

The more $res_k(i)$ is, then the more agent k can exert it's function at node i , therefore at afterwards distribution the more it is prone to migrate to node i .

3.2.2 Apply Ant Algorithms to Realize the Agent One-Hop Migration in the Defense Progress

Ant algorithms is collective intelligence that studies how the actions and inter-relations of a set of simple agents (for example, bees, ants, etc.) carry out global objectives of the system where these agents are immersed[13]. The ant algorithm was firstly used to solve the TSP. According to the ant algorithms, the ant transition rule is mainly decided by the pheromone left by other ants on the path and heuristic value. In the TSP, the more shorter a path is, the more the number of ants that go through the path, then the more the pheromone left by ants. And ants are prone to select the path with more pheromone to travel so as to toward the optimal result.

When the defense system make defense, agents need to migrate. If agent k want to migrate from i to j , we should consider two factors: one is the pheromone on the path (i, j), the more agents go through path (i, j), the more pheromone is; the other is the comparison between $res_k(i)$ and $res_k(j)$.

According to the ant algorithms, the transition rule of agent is:

$$p_{ij}(k) = \begin{cases} \frac{[\tau(i,j)]^\alpha \times [\eta(i,j)]^\beta}{\sum_{u \in \text{ADJ}_k(i)} [\tau(i,u)]^\alpha \times [\eta(i,u)]^\beta} & \text{if } j \in \text{ADJ}_k(i) \text{ and } p_{ij}(k) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where $p_{ij}(k)$ denotes the probability that agent k migrate from i to j ; $\text{ADJ}_k(i)$ denotes the adjacent nodes of i ; $\tau(i, j)$ denotes the pheromone on the path (i, j) ; $\eta(i, j)$ is a heuristic value; α and β are parameters to control the relative influence importance between pheromone and heuristic value on agent migration probability.

Pheromone update formula is shown as follows:

$$\tau_{ij}(n+1) = \rho \tau_{ij}(n) + \Delta \tau_{ij} \quad (6)$$

Where ρ is a parameter, $(1-\rho)$ denotes the waning degree of pheromone from time n to time $n+1$.

$$\Delta \tau_{ij} = \sum_x \Delta \tau_{ij}^x \quad (7)$$

In Equation (7), x denotes the number of agents; $\Delta \tau_{ij}^x$ denotes the pheromone left by agent x on path (i, j) .

$$\Delta \tau_{ij}^x = \begin{cases} \frac{Q}{(\sigma_1 d_{ij} + \sigma_2 m_{ij}^x)}, & \text{if agent } x \text{ passed } (i, j) \text{ in this migration process} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Among those Q is a constant which can be got by experiment; d_{ij} denotes the distance between i and j ; m_{ij}^x denotes the migration cost of agent x from i to j , more detail can be seen in [14]; σ_1 and σ_2 are used to control the relative importance between d_{ij} and m_{ij}^x .

In this paper, considering the factual situation of intrusion detection system, we design the heuristic value as follows:

$$\eta_k(i, j) = res_k(j) - res_k(i) + C_k \quad (9)$$

C_k is a constant number which is decided by experiment. We can see that only if $res_k(i) - res_k(j)$ is more than C_k , then $\eta_k(i, j)$ is negative, therefore migration probability is zero.

According to (8), (11) and (7), then the migration probability of agent k from i to j at time $n+1$ is shown as follows:

$$p_{ij}(k) = \begin{cases} \frac{[\rho\tau_{ij}(n) + \Delta\tau_{ij}]^\alpha \times [res_k(j) - res_k(i) + C_k]^\beta}{\sum_{u \in ADJ_k(i)} [\rho\tau_{iu}(n) + \Delta\tau_{iu}]^\alpha \times [res_k(u) - res_k(i) + C_k]^\beta} & \text{if } (j \in ADJ_k(i)) \text{ and } (p_{ij}(k) \geq 0) \text{ and } (\text{the denominator} \neq 0) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

4 Experiment and Conclusion

We have developed a network defense prototype system based on mobile agent technology. Now we embed the *topology-adapted network defense model* provided by this paper into the prototype system and make simulation test.

In our experiment, we make two kinds of tests: 1). Running the original prototype without the introduction of the model provided by this paper (Method 1); 2). Using our topology-adapted network defense model (Method 2). And we make comparison between the intrusion detection efficiencies of Method 1 and Method 2. The intrusion detection efficiency is defined as the proportion of the number of successful intrusion detection to the total number of simulation tests.

In the simulation experiment, we adopt Expert, an unix tool, to simulate the intrusion. In our experiment, we get different network topology by change the amount of network nodes from 3 to 13. And in the network, we implement full inter-connection among the nodes.

From Fig 3, we can see that: when the nodes number is 3, the comparison is not obvious, the reason is that since the topology isn't so complex that our model can't play it's advantage well; however, with the increase of node number, then the network topology is more complex, the efficiency of the prototype system with our model is higher than the one of the original prototype system, and the comparison is more obvious.

Therefore, the simulation result proves that our model is feasible, especially when the network topology is complex. And our model can adapt itself with the change of network topology.

However, we can see that the defense efficiency is too low to be applied in factual situation application, so next we should improve the efficiency of our prototype system and make it be able to be applied in factual situation.

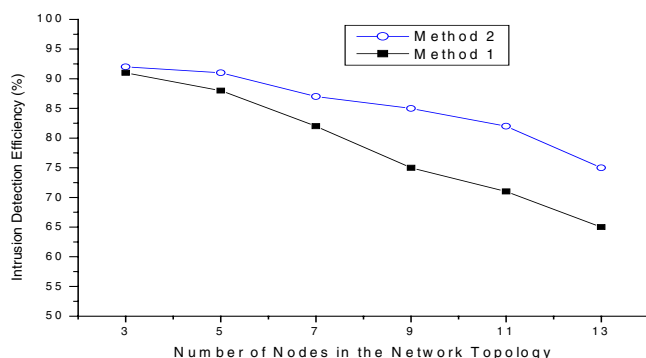


Fig. 3. The Comparison between the Two Scenarios in the Experiment

This paper provides a network defense model adapted to dynamic topology, and makes detail explanation for the architecture and principle of the model. Our future task should focus on the more development of the system and realize a real applied network defense system that can adapt to dynamic topology.

References

1. Wenke Lee, Salvatore J.Stolfo, Kui W.Mok. A Data Mining Framework for Building Intrusion Detection Models. IEEE Symposium on Security and Privacy 1999:120-132.
2. Ludoric ME. GASTATA, A Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis. www.inf.tu-dresden.de/publications/1998/wv-1998-01.ps.gz.
3. Jai Balasubramanian, Jose Omar Garcia-Fernandez, David Isacoff, E.H.Spafford, and Diego Zamboni. An Architecture for Intrusion Detection using Autonomous Agents. Department of Computer Science, Purdue University; Coast TR 98-05, 1998.
4. Asaka M., Okazawa S., Taguchi A., Goco S. A Method of Tracing Intruders by Use of Mobile Agents. INET99, June 1999.
5. Sangeeta Parikh. A Framework of System Integrator for MAIDS[Report for the degree of Master of Science]. Iowa State University, Ames, Iowa. 2001.
6. Yichuan Jiang, Yiping Zhong, Shiyong Zhang. A Network Defense Model Adapted to Variable Topology. In: The Proceeding of Network Research Workshop 2003, APAN. Busan, Korea, Aug 2003.
7. C.A.Carver, Jr. Adaptive Agent-based Intrusion Response. [PH.D Thesis]. Texas A&M University. 2001.
8. Hwa-Chun Lin, Chien-Hsing Wang. Automatic Topology Discovery Using Mobile Agents. In: Proceeding of the International Workshop on Agent Technologies over Internet Applications, September 26 -- 28, 2001.
9. Bruce Lowekamp, David R.O'Hallaron, Thomas R.Gross. Topology Discovery for Large Ethernet Networks. SIGCOMM'01, August 27-31, 2001.
10. Holland, John H. Genetic Algorithms. Scientific American, 267 (July 1992), 60-78. [mc95]M.Crosbie, G.Spafford. Defending a Computer System Using Autonomous Agents. In: Proceedings of the 18th National Information Security Conference, October 1995.

11. Chris Herringshaw. Detecting Attacks on Networks. IEEE Computer Magazine, Dec. 1997: 16-17.
12. P.D'haeseleer, S.Forrest, and P.Helman. An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications. In Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy (1996): 110-120.
13. Colorni A, Dorigo M and Maniezzo V. Distributed Optimization by Ant Colonies [A]. In Proceedings of the 1st European Conf. Artificial Life [C], Pans, France: Elsevier, 1991, 134-142.
14. W.Jansen, P.Mell, T.Kargiannis, D.Marks. Mobile Agents in Intrusion Detection and Response. In Proceedings of the 12th Annual Canadian Information Technology Security Symposium, Ottawa, Canada, June 2000.
15. Sapon Tanachaiwiwat, Kai Hwang, and Yue Chen. Adaptive Intrusion Response to Minimize Risk over Multiple Network Attacks. ACM Trans on Information and System Security, August 19, 2002.
16. Jose Aguilar. A General Ant Colony Model to Solve Combinational Optimization Problems. REVISTA COLOMBIANA DE COMPUTACION, Volumen 2, Number 1, pages:7-18.
17. HUA Si-Yun. Stochastic Processes. South-East University Pree, 1988. 49-75.
18. LIN-Chuang. Computer Network and Computer System Performance Evaluation. Beijing:Tsinghua University Press, 2001, 176-185.
19. HE-Yong, WANG Hong-Wei, FEI-Qi. Petri-Net-Based Modeling Support. Journal of Huazhong University of Science & Technology, 1995, Vol.23, No.5: 52-56.

AT-RBAC: An Authentication Trustworthiness-Based RBAC Model*

Lunwei Wang, Lifeng Wei, Xiangke Liao, and Huaimin Wang

School of Computer Science, National University of Defense Technology
Changsha 410073, China
wang_lunwei@sina.com

Abstract. In current operating systems, the strength of authentication mechanism does not work on the authorization of the user, which leaves the system security compromise that the user who has passed weak authentication mechanism may have many access rights. This paper firstly puts forwards the thought of authentication trustworthiness, the aim is to give each authenticated user his authentication trustworthiness. According to user's trustworthiness, the system will decide which access rights he will have. The more strength is the authentication mechanism, the larger is the user's authentication trustworthiness. The user's authentication trustworthiness will be taken as one of access control decision elements, so as to prevent the user with less trustworthiness from owning many access rights. Based on the authentication trustworthiness, this paper puts forwards the authentication trustworthiness-based *RBAC* model. The model associates authentication trustworthiness with *RBAC* model, and the authentication trustworthiness of the authenticated user will be decision information to activate his roles and permissions, only those users who satisfy role trust activation condition can activate their roles, users who satisfy permission trust activation condition can activate their permissions. The model provides trust authorization by user's role and permissions trust activation, satisfies the requirement that different authentication mechanisms with different strength will correspond to different access rights.

1 Introduction

Currently, many operating systems implement authentication mechanisms with Plugable Authentication Module, denoted by *PAM*^[1]. With the *PAM* framework, multiple authentication technologies can be added without changing any of the system entry services such as login, thereby preserving existing system environments. *PAM* framework provides great flexibility to implement and apply newest authentication technologies, but at the same time, leaves the system security compromise that the user who has passed weak authentication mechanism may have many access rights, even the administrator rights. The main reason is that the strength of authentication mechanism does not work on the user's authorization.

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2002AA1Z2101, "Operating System kernel for Server".

To upper problem, this paper firstly puts forwards the thought of authentication trustworthiness, which reflects the degree of trustworthy of the user who has passed system authentication. Then, this paper puts forwards authentication trustworthiness-based *RBAC* model, denoted by *AT-RBAC*, which associates authentication trustworthiness with *RBAC* model, and takes the user's authentication trustworthiness as the decision condition to activate his roles and permissions. Only those users who satisfy role trust activation condition can activate their roles, users who satisfy permission trust activation condition can activate their permissions. The model keeps the advantage of permission management, mainly emphasizes on the trust activation of user-role and role-permission, so that the strength of authentication mechanism corresponds to access rights of authenticated users.

The remainder of this paper is organized as follows. Section 2 gives a summary of *RBAC96* model. Section 3 puts forwards the thought of authentication trustworthiness and trust access condition. Section 4 puts forwards the *AT-RBAC* model, and describes the role trust activation and permission trust activation condition. Finally, section 5 provides a summary of the paper.

2 *RBAC96* Model

Among some *RBAC* models proposed in [2,3,4], *RBAC96* model proposed by Sandhu et al has been accepted widely. The *AT-RBAC* model is also based on *RBAC96*. Now, we at first give a summary of *RBAC96* model.

The elements of *RBAC96* model are:

Definition 1 (*RBAC* Structure)

U is a set of users, for example $\{u_1, u_2, \dots, u_n\}$.

R is a set of roles, for example $\{r_1, \dots, r_m\}$.

P is a set of permissions, for example $\{p_1, \dots, p_q\}$.

S is a set of sessions, for example $\{s_1, \dots, s_t\}$.

$UA \subseteq U \times R$ is a many-to-many user-to-role assignment relation.

$PA \subseteq P \times R$ is a many-to-many permission-to-role assignment relation.

$SA \subseteq S \times U$ is a session-to-user relation.

$RH \subseteq R \times R$ is a partial ordered role hierarchy (written as $\triangleright$ in infix notation)

Definition 2 (*RBAC* Global Functions)

$role\_set: U \cup P \cup S \rightarrow 2^R$

$u \in U: role\_set(u) = \{r \in R \mid (u, r) \in UA\}$

$p \in P: role\_set(p) = \{r \in R \mid (p, r) \in PA\}$

$s \in S: role\_set(s) = \{r \in R \mid (user(s), r) \in UA\}$

$user\_set: R \rightarrow 2^U: user\_set(r) = \{u \mid (u, r) \in UA\}$

$user: S \rightarrow U: user(s) = u, (s, u) \in SA$

$perm\_set: R \rightarrow 2^P: perm\_set(r) = \{p \mid (p, r) \in PA\}$

There is a collection of constraints that determine whether or not values of various components of the *RBAC96* model are acceptable (only acceptable values will be permitted).

3 Authentication Trustworthiness

3.1 Basic Concepts of Authentication Trustworthiness

Because authentication and authorization are disjointed, the strength of authentication mechanism does not work on the authorization of the user, this paper puts forward the thought of authentication trustworthiness, which reflects the degree of authenticated user's trustworthy. The user's authentication trustworthiness will be taken as a basic qualification, when judging access request, the system should make an access control decision based on the user's authentication trustworthiness. The authentication trustworthiness builds a bridge between authentication and access control.

Before defining some concepts of authentication trustworthiness, we should at first give the definitions of subject and object, so as to describe authentication trustworthiness more precisely.

Definition 3 Subject is a user entity which can initiate action, such as user processes. Object is a passive subject action undertaker, such as data, file, etc. The sets of subject and object can be denoted separately by S and O .

The definitions relative to authentication trustworthiness are:

Definition 4 Authentication Trustworthiness reflects the degree of trustworthiness of the subject who has passed system authentication, denoted by $AT(s)$. The value of $AT(s)$ is between 0 and 1. The larger is the value, the more is the degree of trustworthiness.

Definition 5 Object Access Trustworthiness represents the least authentication trustworthiness of the subject who can access object, denoted by $AT(o)$. The value of $AT(o)$ is between 0 and 1. The larger is the value, the more authentication trustworthiness of the subject is needed.

Definition 6

$f_{sub} : S \rightarrow [0, 1]$ represents authentication trustworthiness functions of the subjects, for example, the authentication trustworthiness functions of subject s is $f_{sub}(s)$.

$f_{obj} : O \rightarrow [0, 1]$ represents access trustworthiness functions of the objects, for example, the access trustworthiness functions of object o is $f_{obj}(o)$.

3.2 Trust Access Condition

Before we define trust access condition, we describe distrust access at first.

Distrust represents the uncertainty of the user's identity. Although the user has passed system authentication, we can not make certain whether he is trusted or not. There are some uncertainties in authentication systems, such as the uncertainties of the authentication mechanisms, authentication rules and authentication conclusions.

The uncertainties of authentication mechanisms show that we can not make certain the reliability and security of the authentication mechanisms completely. Are the authentication mechanisms secure and correct? Is there a trusted path to ensure that the transferring authentication information can arrive at correct authenticator? Is there a Trojan horse program to cheat the user?

The uncertainties of the authentication rules show that if we say the authenticated user is trusted, it is just likelihood. The administrator should have certain untrustworthiness to the authentication rules.

The uncertainties of the authentication conclusions show that the precondition includes several kinds of uncertainties, after using uncertain rules, the conclusions have uncertainties inevitably.

So, we expect the distrust subjects do not be allowed to access system objects.

By analyzing the distrust access, we can know that the trust access condition should be defined that preventing the subject from accessing object under the condition the authentication trustworthiness of the subject is less than the access trustworthiness of the object.

Theorem 1 Trust Access Condition $(s,o) \in (S \times O)$ satisfies trust access condition iff $f_{\text{sub}}(s) \geq f_{\text{obj}}(o)$.

4 Authentication Trustworthiness-Based RBAC

4.1 Introduction

AT-RBAC model associates authentication with access control, just as fig 1. Authenticated user will gain his authentication trustworthiness, which is taken as access control decision information. In *RBAC96* model, the user does not access system resources directly, through a certain role, according to his permissions, he knows whether he can access the system resources or not. So, *AT-RBAC* model takes roles and permissions as objects. Object access corresponds to role activation and permission activation, object trust access condition corresponds to role trust activation condition and permission trust activation condition. So we have these definitions:

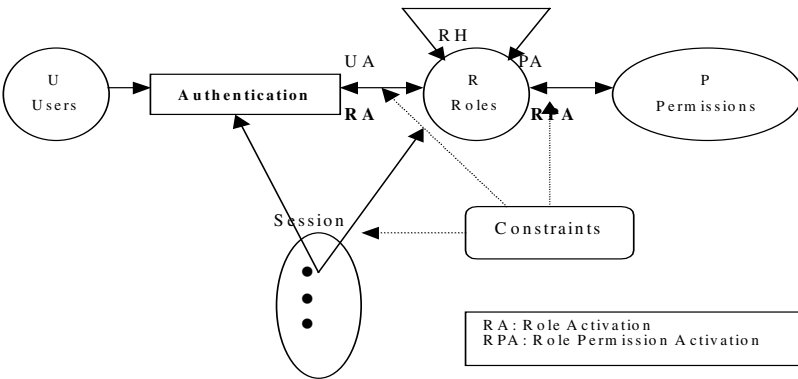


Fig. 1. *AT-RBAC* model

Definition 7 Role Activation Trustworthiness represents that the user's authentication trustworthiness must exceed this value so that he can activate his role. This value,

denoted by $AT(r)$, is between 0 and 1. The more is the value, the more the authentication trustworthiness of the user who wants to activate this role is needed.

One user can correspond to several roles, so we should configure every role his activation trustworthiness.

By Theorem 1, we can easily get the lemma of role trust activation condition:

Lemma 1 Role trust activation condition $(u, r) \in (U \times R)$ satisfies the role trust activation condition iff $f_{\text{sub}}(u) \geq f_{\text{obj}}(r)$.

In *AT-RBAC* model, authentication trustworthiness not only embodies in the *UA* component, but also *PA* component. On *PA* component, *AT-RBAC* model takes permission as object. So we have this definition:

Definition 8 Permission Activation Trustworthiness represents that the user's authentication trustworthiness must exceed this value so that he can activate his permission. This value, denoted by $AT(p)$, is between 0 and 1. The more is the value, the more the authentication trustworthiness of the user who wants to activate this permission is needed.

One role corresponds to several permissions, so we should configure the permission his activation trustworthiness.

By Theorem 1, we can easily get the lemma of permission trust activation condition:

Lemma 2 Permission Trust Activation Condition $(u, p) \in (U \times P)$ satisfies the permission trust activation condition iff $f_{\text{sub}}(u) \geq f_{\text{obj}}(p)$.

The permission activation trustworthiness in *AT-RBAC* model is configured by the administrator, role activation trustworthiness is calculated according to the permission activation trustworthiness. $\forall r \in R, AT(r) = \min \{ \bigcup_{p \in \text{perm\_set}(r)} AT(p) \}$ represents that

the activation trustworthiness of role r is the least value of his permission activation trustworthiness, in which the permissions belong to role r .

AT-RBAC model emphasizes on the role and permission trust activation, demands all the roles activation should satisfy role trust activation condition, and all the permission activation should satisfy permission trust activation condition. Based on *RBAC96* model, *AT-RBAC* model implements two level constraints on access: role activation constraint and permission activation constraint.

4.2 AT-RBAC Structure

AT-RBAC model inherits all the element of *RBAC96* model, and extends the *RBAC96* model.

Definition 9 (AT-RBAC Structure)

In *AT-RBAC* structure, the definition of U , R , P , S are the same as the definition in *RBAC96* model. We define $AT(U)$, $AT(R)$ and $AT(P)$, in which, $AT(U)$ is the set of authentication trustworthiness of authenticated users, $AT(P)$ is the set of permission activation trustworthiness configured by the administrator, $AT(R)$ is the set of role activation trustworthiness, $AT(R) = \bigcup_{r \in R} AT(r)$.

$UA \subseteq U \times R$, $PA \subseteq P \times R$, $SA \subseteq S \times U$, $RH \subseteq R \times R$ are the same as the definitions in *RBAC96* model.

$RA \subseteq U \times R$ is a many-to-many user-to-role activation relation, which reflects the active role set of current user. The active role set is denoted by AR , for example, $AR(u)$.

$RPA \subseteq R \times P$ is a many-to-many role-to-permission activation relation, which reflects the active permissions set of current role. The active permissions set is denoted by AP , for example, $AP(r)$.

Definition 10 (AT-RBAC Global Functions)

In *RBAC96* model, we have defined *role_set*, *user_set*, *user* and *perm_set* functions, *AT-RBAC* keeps these global functions, and extends some global functions on RA and RPA .

role_activate: $U \cup S \rightarrow 2^R$:

$$u \in U : \text{role\_activate}(u) = \{r \mid (u, r) \in UA \wedge AT(u) \geq AT(r)\} = AR(u)$$

$$s \in S : \text{role\_activate}(s) = \{r \mid (user(s), r) \in UA \wedge AT(user(s)) \geq AT(r)\}$$

perm_activate: $U \cup S \rightarrow 2^P$:

$$u \in U : \text{perm\_activate}(u) = \bigcup_{r \in AR(u)} \{p \mid (r, p) \in PA \wedge AT(u) \geq AT(p)\}$$

$$\begin{aligned} s \in S : \text{perm\_activate}(s) \\ = \bigcup_{r \in AR(user(s))} \{p \mid (r, p) \in PA \wedge AT(user(s)) \geq AT(p)\} \end{aligned}$$

4.3 Role Trust Activation and Permission Trust Activation

Because *RBAC96* model emphasizes on access control, the premise is that the authentication mechanism is secure enough to ensure that the authenticated user is a correct and trusted user, not an illegal user. That is to say, the authenticated user has enough trustworthiness. But there are some uncertainties in authentication systems, we can not ensure that the authenticated user is trusted. From security aspect, authenticated user with little trustworthiness can not be authorized all his rights. *AT-RBAC* model make a bridge between authentication and access control through authentication trustworthiness. The users with different authentication trustworthiness can activate their different roles. In UA component of the model, the users are taken as subjects, assigned roles are taken as objects, the attribute of the subjects are their authentication trustworthiness, the attribute of the objects are their role activation trustworthiness, role trust activation condition is the secure policy. So, role trust activation can be seen as a simple access control model.

In *AT-RBAC* model, role corresponds to assigned permissions. Because different authentication mechanisms have different strength, the authenticated users who have

different authentication trustworthiness should have different permissions. In *PA* component of *AT-RBAC* model, the user is taken as subject, the permissions of active role are taken as objects, the attribute of subject is his authentication trustworthiness, the attribute of object is his permission activation trustworthiness, and the permission trust activation condition is the secure policy. So, permission trust activation is similar to the role trust activation, and also can be seen as a simple access control model.

Authenticated user who logs in system has an active role, according his authentication trustworthiness, when activating permissions, by permission trust activation condition, the *ADF* can decide whether he can activate the permissions or not. Permission trust activation is a kind of access control constraint between user and permissions.

4.4 Analyse of the Model

The most frequently mentioned constraint in the context of *RBAC96* is mutually exclusive roles. The same user can be assigned to at most one role in a mutually exclusive set. This supports separation of duties.

There are other kinds of constraints in *RBAC96* model, such as cardinality constraint and time constraint [6]. Cardinality constraint refers to that a role can have a maximum number of members.

AT-RBAC model applies access control constraint to role activating. If the user can activate his role r , his authentication trustworthiness must not be less than the role activation trustworthiness, and satisfy role trust activation condition. At the same time, the model applies access control constraint to permission activating. If the user can activate his permission p , his authentication trustworthiness must not be less than the permission activation trustworthiness, and satisfy permission trust activation condition.

Based on authentication trustworthiness, *AT-RBAC* model builds two level access control constraints to extend *RBAC96* model. From above description, we can see that *AT-RBAC* model keeps the advantages of *RBAC96* model, and at the same time, by trustworthiness constraint, ensures that the roles and permissions can be trust activation, so as to satisfy the real requirement, and solve the disassociation problem between authentication and access control.

5 Summary

This paper firstly puts forwards the thought of authentication trustworthiness. Based on the authentication trustworthiness, this paper puts forwards the *AT-RBAC* model. The model associates authentication trustworthiness with *RBAC* model, and the authentication trustworthiness of the authenticated user will be decision information to activate his roles and permissions, only those users who satisfy role trust activation condition can activate their roles, users who satisfy permission trust activation condition can activate their permissions. The model can provide trust authorization by user's role and permissions trust activation, so as to satisfy the requirement that different authentication mechanisms with different strength will correspond to different access rights.

AT-RBAC model constrains the ability to activate his roles by the user's authentication trustworthiness and role activation trustworthiness, and constrains the ability to activate his permissions by the user's authentication trustworthiness and permission activation trustworthiness, so as to prevent the user with little authentication trustworthiness authenticated by weak authentication mechanisms from obtaining larger activation trustworthiness of roles and permissions.

References

1. V Samar, C Lai. Making login services independent of authentication technologies. Sun Microsystems, 1995, <http://java.sun.com/security/jaas/doc/pam.html>
2. Sandu R S, et al. Role-Based Access Control Models. *IEEE computer*, 1996, 29(2): 38~47
3. Sandu R S, et al. Role-based Access Control: A Multi-Dimension Vies. In: *Proc. Of the 10th Annual Conf. On Computer Security Applications*. 1994
4. Ferraiolo DF, Sandhu R, Gavrila S. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 2001, 4(3): 224~274.
5. AHN G-J, Sandhu R. Role-Based authorization constraints specification. *ACM Transactions on Information and System Security*, 2000, 3(4): 207~226.
6. Dong GY, Qing SH, Liu KL. Role-based authorization constraint with time character. *Journal of Software*, 2002, 13(8): 1521~1527
7. Jaeger, T. On the increasing importance of constraints. In: *Proceedings of 4th ACM Workshop on Role-Based Access Control*. Fairfax, VA: ACM Press, 1999. 33~42.
8. Chen, Fang, Sandhu, R. Constraints for role-based access control. In: *Proceedings of the ACM RBAC Workshop*. MD: ACM Press, 1996. 39~46.

WBEM Based Distributed Network Monitoring*

Bo Liu and Hui Liu

School of Computer, National University of Defense Technology
410073, Changsha, Hunan, P.R.China
bo.liu@263.net

Abstract. In the paper we identify the needs in efficient management of distributed networks and some problematic areas in the field. Then we introduce Web Based Enterprise Management (WBEM) to address the problem of providing a unified way to model all kinds of managed elements in a single information model in heterogeneous network environments. The advantages brought about by the use of WBEM in network management solve some critical problems existing in current network management. Based on the brief description of components of WBEM, we discuss in depth the basic WBEM instrumentation and multi-tiered WBEM enabled management infrastructure. We also apply this multi-tiered management infrastructure to a network management application scenario to monitor network activities for unexpected behaviors.

Keywords: Network Security, Network Monitoring, Web Based Enterprise Management, Common Information Model

1 Introduction

The field of network management is of strategic importance to modern computer networks. The purpose of network management is to maintain networked systems availability or improve their performance. There are usually two aspects in network management: monitoring and control. Network monitoring is commonly regarded as gathering information of network and/or networked systems, and representing them in an effective way. On the other hand, network control is responsible for taking action to network and system tuning.

Unfortunately, network management is also a field that is fraught with intricacies and problems which are described in [1], [2]. To provide a unified way to manage heterogeneous networks, the International Telecommunications Union (ITU) has proposed a network management model aimed at understanding the major functions of network management. Five conceptual areas involved with the model include performance management, configuration management, accounting management, fault management and security management [3].

These conceptual areas are useful in understanding the goals of network management and monitoring. In the paper, the term “network monitoring” mainly refers to the behaviors of observing the running state of the network, gathering information of networked systems, capturing the kinds of events occurred in the network, with ade-

* Supported by the National High-Technology (863) Program under Grant 2003AA142080.

quate accuracy and reasonable latency. During the course of network monitoring, no any corrective actions will be taken. Another term "network management" will be used to refer to the behaviors that will both monitor the network and take corrective or preventative maintenance actions. Therefore, network monitoring is a subset of network management, so a great deal of network management concepts, ideas and operation manners will be applicable to network monitoring.

In following sections, we will discuss both network monitoring and network management, but we will pay more attentions to network monitoring in our work because obviously network management should make its decisions basing upon accurate and reliable information provided by network monitoring. The information could be used for resource management, scheduling applications, providing performance information to network-aware applications, etc.

The remainder of the paper is organized as follows. In Section 2, we summarize related works. In Section 3 we make a brief introduction to components of WBEM. In Section 4 we discuss the WBEM enabled management infrastructure. In Section 5, we apply the multi-tiered WBEM architecture to build a scalable network monitoring application. Future works are discussed in Section 6.

2 Related Works

Currently, the architecture of Network Management (NM) often includes a management application (manager) and the managed entities (agents) which are embedded within Network Elements (NEs). Management interactions make use of the Client/Server approaches, with the manager collecting status data and taking control actions through the agents. The communication between the manager and the agents is facilitated by kinds of NM protocols such as the Simple Network Management Protocol (SNMP) [4], the Common Management Information Protocol (CMIP) [5] and Common Object Request Broker Architecture (CORBA) [6]. Within these protocols, abstractions of physical elements in the network are represented by different managed objects.

The existence of number of potential conflicting standards with no common models causes the managed elements could not be compatible with each other, so network administrators have to use multiple different management applications to manage the networks. Furthermore, though already used widely, some NM protocols, such as SNMP, have been known to have several remarkable disadvantages [7]. This situation has made a great need for a standard that can unify the current standards while also be able to model all thinkable forms of NEs by a single information model. An attempt to achieve this goal is Web Based Enterprise Management (WBEM).

WBEM specification is an ongoing initiative started by the Distributed Management Task Force (DMTF) which consists of a large number of companies involved in the network management scene, such as Sun, Microsoft, Cisco, Compaq, Intel, 3Com and over 70 others. The specification defines management architecture, management protocol, management schema, and object manager. This initiative proposes a common method of managing enterprise systems without requiring a complete overhaul of the existing management architecture. By utilizing the WBEM uniform model any management source can be accessed in a common way. Some advantages that can be achieved by network management using WBEM could be found in [8].

A large number of equipment vendors have started to release products supporting WBEM. Examples of management systems capable of managing WBEM elements are CiscoWorks and BMC patrol. Microsoft System Management System (SMS) uses WBEM for management information as well as many other components of Windows NT. Microsoft also has implemented WBEM in its Windows Management Interface (WMI) as well as its Common Information Model (CIM). Compaq's Tru64 UNIX on Alpha Server is another example of an operating system capable of being managed with WBEM. Besides these, there are also four different Open Source CIM Object Managers (CIMOMs) available at the moment. These CIMOMs include openCIMOM, Pegasus, OpenWBEM and WBEMServices.

3 Components of WBEM

Partial contents in the section are excerpted from [9]. It is put here for easy reference. Refer to [10], [11] for details about WBEM and CIM.

As shown in Fig.1, The DMTF has developed a core set of standards that make up WBEM, which includes a data model, the CIM standard; an encoding specification, xmlCIM Encoding Specification; and a transport mechanism, CIM Operations over HTTP.

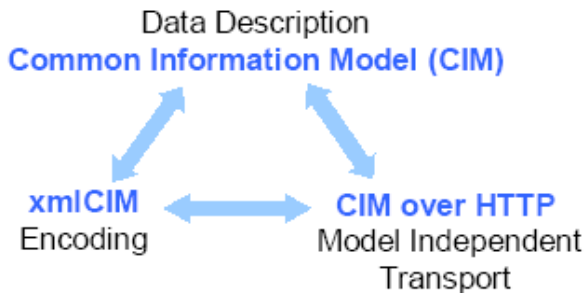


Fig. 1. The components of WBEM and the relationships among these components

The CIM is the language and methodology for describing management data. CIM schema maintained by the DMTF includes models for Systems, Applications, Networks (LAN) and Devices. The CIM schema will enable applications from different developers on different platforms to describe management data in a standard format so that it can be shared among a variety of management applications. The xmlCIM Encoding Specification defines XML elements, written in DTD, which can be used to represent CIM classes and instances. The CIM Operations over HTTP specification defines a mapping of CIM operations onto HTTP that allows implementations of CIM to interoperate in an open, standardized manner and completes the technologies that support WBEM.

The core of the WBEM is a data modeling concept plus a set of data models referred to CIM Schemas which are suited for management purposes. To administrators, it is possible to create extensions to the CIM Schemas to address the respective aspects of management. So the major advantage of WBEM based network management is the capability of one WBEM client application to remotely manage all different kinds of WBEM enabled platforms.

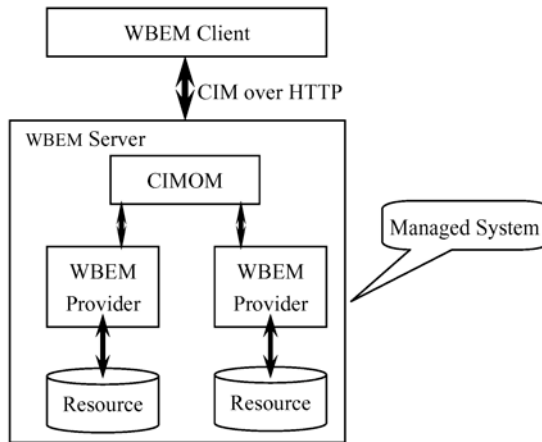


Fig. 2. The basic WBEM instrumentation

4 WBEM Enabled Management Infrastructure

To construct a management system capable of WBEM there needs to be a WBEM client and at least one WBEM server, as shown in Fig. 2. The WBEM server consists of some WBEM providers which usually are called agents in other NM architectures, a CIMOM, and an interface between WBEM client and WBEM server. Instead of accessing the providers directly, the client forwards requests to the CIMOM in the WBEM server, which in turn delegate these requests to providers. This structure makes it so that the CIMOM is responsible for the communication between the client and server, manages the schema and routes requests down to the providers, while the providers will act as plugins to the CIMOM and implement the link between the CIMOM and the software entities that are responsible for handling the underlying managed objects. We usually call this model of giving WBEM access to the managed resources as “basic WBEM instrumentation”.

The basic WBEM instrumentation mentioned above is well-suited for management scenarios in a relatively small network environment. However, if we want to apply the basic WBEM instrumentation to enterprise network management, considering the distributed network environments with hundreds or thousands of systems and much more kinds of associated management objects, the WBEM client itself may become the bottleneck of the entire management architecture.

To address these problems, by introducing so called Intermediate Level Management Server (ILMS) into basic WBEM instrumentation, we have adopted the multi-tiered management architecture [12] to build scalable management infrastructure for distributed network monitoring. The ILMSes will concentrate and consolidate information from systems and resources to be managed. As shown in Fig. 3, actually the ILMSes are CIMOMs which are not dedicated to a certain system or resources, but maintain management information from many systems and resources to be managed. In fact, to those providers primarily dealing with elements to be managed, the providers in ILMS will act as management application to control resources on the managed systems.

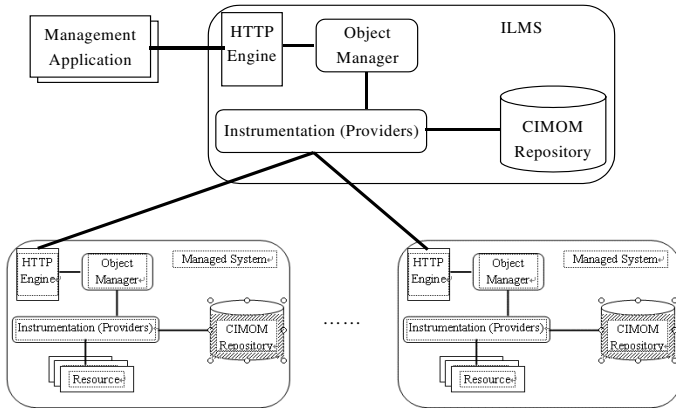


Fig. 3. The Multi-Tiered management architecture

5 A Network Monitoring Scenario Enabled by WBEM

Monitoring variable network messages as they traverse in the network will provide the capability to identify intrusive activity at the time it is occurring or soon after. By catching suspicious network activity, we can immediately begin to investigate the activity and minimize the impacts of the possible damages caused by these intrusions [13].

We are applying the multi-tiered WBEM management infrastructure discussed in previous section to monitor an Intranet which consists of about 20 subnets and over 400 hosts. The main object of our work is to inspect network activities for unexpected behaviors via efficiently network monitoring. Based on the accurate network topology, we plan to deploy agents, i.e. providers to important network segments and critical hosts to collect network and system information.

As shown in Fig. 4, all network elements to be monitored will have management instrumentations and providers to be deployed. A top Level ILMS will be used in the architecture to enable monitoring application to use one single point to achieve the network monitoring efficiently. In the Figure, the connecting lines among ILMSes and CIMOMs carry normal WBEM protocols and CIM operations to delegate monitoring requests from the monitoring application to the applicable subordinate ILMSes or CIMOMs. If necessary, the dotted lines can be used by the monitoring application to access the CIMOMs residing in the managed systems or ILMSes in lower levels to query network and system information directly. But then this operation will require detailed network topology to be known by monitoring application.

To comply with WBEM specification, we develop and deploy four different types of provider interfaces to enable WBEM based network monitoring. Each provider is a software component that provides information about a logical or physical entity to CIMOM.

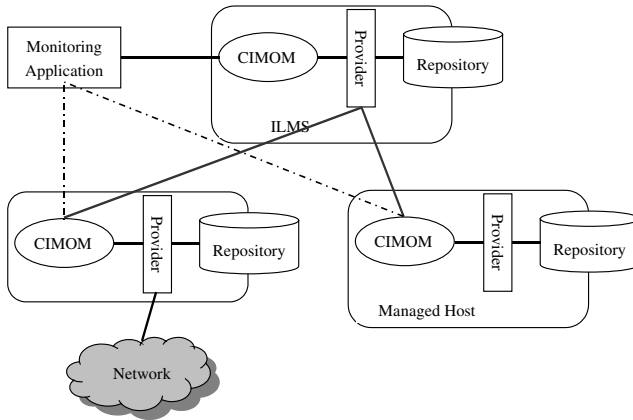


Fig. 4. Network monitoring Architecture enabled by WBEM

1. Instance provider interface is responsible for enumerating the available resource entities and performing actions on these instances, providing utility methods to support instances of specific CIM classes.
2. Method provider interface lists methods that a provider supports to manage the resources and executes extrinsic methods on a certain class or instance.
3. Property provider interface supports retrieval and modification of CIM properties, such as get and set methods to modify name-value pair of a CIM instance.
4. The indication provider interface allows a client to subscribe or unsubscribe to certain events. When an event is triggered within the CIMOM, it can be exported in multiple ways based on the indication provider that are loaded.

Because the logs of network traffic may contain evidence of suspicious or unexpected network activities, we may be able to identify intruder reconnaissance in advance of an intrusion, or the attempted or successful intrusions soon after they have occurred by inspecting or analyzing these log files efficiently. In our network monitoring scenario, data about network activities will be collected by a number of providers from a variety of sources such as

- log files of routers, firewalls, hosts and other network devices
- network alert and error reports by other network monitoring tools
- network performance statistics
- probes including ICMP pings, port probes, SNMP queries, and so on

By analyzing these collected data, the monitoring application will try to identify unexpected or suspicious network behaviors which include

- unexpected changes in network performance
- abnormal network traffic
- non-standard or malformed packets
- unauthorized scans and probes, and other intrusive activities already known by the monitoring application

6 Future Work

The fields of network monitoring and network management are large, and a great deal of scope exists for research within these fields. There are three obvious extensions to the work that has been covered in the paper. Of course, there is also scope for work outside of the confines in the paper.

1. Information analysis in-depth and decision making. There are a whole lot of possibilities that can be explored in this area. Many approaches and methods can be borrowed from other research communities, such as AI, to make our analysis and decision more accurately. The purpose of this research is to help reviewing and investigating network error reports, network performance statistics, notifications from network-specific alert reports and anything that appears anomalous, and identifying any unexpected or suspicious network behaviors.
2. Within the WBEM enabled monitoring infrastructure, improving the extensions to CIM schema to adapt the requirements of inspecting network activities for unexpected behaviors.
3. Utilizing the result of network topology discovery more efficiently to deployed the kinds of management instrumentations so as to improve the accuracy and reliability of network monitoring.

References

1. Guy Antony Halse, "Novel Approaches to the Monitoring of Computer Networks", available at <http://mombe.org/~guy/papers/msc-2003/html>.
2. Jacob W. Kallman, Pedrum Minaie, Jason Truppi, Sergiu M. Dascalu, Frederick C. Harris, Jr., "Software Modeling for Open Distributed Network Monitoring", available at <http://www.cs.unr.edu/~fredh/papers/conf/032-smotodnms/paper.pdf>.
3. Marshall T. Rose, "The Simple Book: An introduction to management of TCP/IP-based internets", Prentice-Hall, 1991.
4. J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," IETF, RFC 1157, 1990.
5. ITU-T, "Information Technology-Open Systems Interconnection-Common Management Information Protocol Definition," Geneva, ITU Recommendation X.711, 1992.
6. Object Management Group, "The CORBA/IIOP Specification 3.0.2 (formal/2002-12-02)", available at <http://www.omg.org/cgi-bin/apps/doc?formal/02-12-02.pdf>.
7. Damianos Gavalas, Dominic Greenwood, Mohammed Ghanbari, Mike O'Mahony, "A Progressive Network Management Architecture Enabled By Java Technology", Proceedings of the 1st International Conference and Exhibition on the Practical Applications of Java, London, UK, April 21-23, 1999.
8. Eun-Ju Ha, Su-Hyung Jo and Jong-Tae Park, "Design and Implementation of Web-based Internet/Intranet Application Service Management System", available at <http://ain.knu.ac.kr/Publication/pdfdata/ConferenceWorkshopProceedingInternational/ejha.pdf>.
9. SNIA Storage Management Initiative, "CIM/WBEM Technology Backgrounder", available at http://www.snia.org/tech_activities/smi.
10. Microsoft, "the Learn WBEM course", available at <http://msdn.microsoft.com/developer/sdk/wbemsdk/default.htm>.
11. Distributed Management Task Force, <http://www.dmtf.org>
12. Viktor Mihajlovski, Adrian Schuur, "CIM Based Distributed Monitoring", available at <http://www-124.ibm.cim/sblim/doc/cimdm.pdf>.
13. Cert Coordination Center, "Monitor and inspect network activities for unexpected behavior", available at <http://www.cert.org/security-improvement/practices/p094.html>.

Multiparty Joint Authentication: Extending the Semantics of Single Sign-On for Grids*

Hui Liu and Minglu Li

Department of Computer Science and Engineering
Shanghai Jiaotong University, 200030 Shanghai, China
{liuhui, li-ml}@cs.sjtu.edu.cn

Abstract. This paper regards Single Sign-On as an accumulation of a series of two-party authentication, multiparty authentication and authorization. Such a comprehension brings new semantics for Single Sign-On in grids: authentication service and authorization service are separable and could communicate with each other through SAML assertions; Single Sign-On could support both two-party and multiparty authentication. Multiparty Joint Authentication (MJA) is designed to simplify multiparty authentication in some security context. This paper describes MJA with graph theory model and proposes its definition formally. The internal sequence diagram of MJA, possible assertion format of MJA, and MJA's interactions with other OGSA services are also illustrated to reveal a systematic view of this paradigm.

1 Introduction

Security solution for grids is just like Achilles' heel. From certain angle, such issues exist through whole lifecycle of designing, implementing, deploying, exploiting and managing any kind of grid systems, and security components are fundamental building blocks to bring grids into reality [1]. On the other hand, current GSI has its intrinsic flaws: proxy certificates may derive a vulnerable trust chains, where compromising of any proxy (child-proxy) acting on behalf of the user (parent-proxy) would destroy believes of the original user as a whole and result in trust crisis. Therefore, IETF has rejected such proxy certificates from the X.509 operating standards [2].

Single Sign-On (SSO) is a primary security requirement for grids [3]. However, SSO is a moving target in different context. For example, two main approaches for SSO are scripting and ticketing; functionalities of SSO must address requirements of supporting multiple authentication mechanisms, simple integration, administration and configuration, flexible policies considerations, etc [4]. Different strategies are always having different influences on downstream activities, i.e., authorizing, scheduling, allocating, accounting, auditing, etc.

* This research is supported by the National Grand Fundamental Research 973 Program of China (No.2002CB312002), and the Grand Project (No.03dz15027) of the Science and Technology Commission of Shanghai Municipality.

This paper presents some new semantics for SSO in grids, which depend on traditional two-party mutual authentication and adopt a novel strategy: firstly, finding an equivalent Multiparty Joint Authentication (MJA) among all involved principals as a whole instead of just mutual authenticating each pair of principals in turn; secondly, issuing an XML Signature based SAML assertion to perform downstream activities. Such a strategy outlines a new vision to improve SSO and leverage its downstream security activities.

The rest of this paper is organized as follows. Section 2 presents related work about SSO. New Semantics for SSO in grids are extended in section 3. Section 4 introduces the mathematical model of multiparty authentication and the definition of MJA. In section 5, a general MJA scenario is illustrated with its internal sequence diagram and possible assertion format. Section 6 indicates MJA might be a component in OGSA security model and Section 7 concludes this paper.

2 Related Work

Single Sign-On (SSO) for general distributed system could be concrete instantiation of some abstract SSO models, such as Broker Authentication Model, Agent-based Broker Authentication Model, Gateway Authentication Model, Script Authentication Model, etc [5]. SSO of GSI is a combination of these models, whose essentials are X.509 certificates, proxy credentials, online credential repository, etc [6, 7].

In order to authenticate users once and access distributed resources without re-authentication, SSO must add additional layer (middleware) to the existing applications. No matter what credential mechanism was employed in the underlying system, X.509 certificates are well-accepted techniques, flexible enough to construct uniform, light-weight, compatible trust model across multiple security domains. Therefore, GSI and other security solutions, e.g., eTrust, Keberos, KSignPassOne, often offer SSO by using X.509 certificates as global, universal, inter-domain credentials.

GSI employs proxy certificates and dynamic delegation to implement SSO in grids. During the creation of proxy certificates, each principal that owns a valid X.509 certificate could act as a potential proxy issuer, whose responsibility is to issue short-lived public certificate that could be used as credential on proxy issuer's behalf. Delegation is very similar to proxy certificates creation, the difference is that the creation occurs over a GSI-authenticated connection, with the result being the remote process acquiring proxy credentials for the user [6].

On line credential repository allow user delegating a set of proxy credentials to the server along with authentication information and retrieval restrictions. At a later time, delegated proxy from the repository may be retrieved and used as any other proxy credentials generated by the user to initiate actions on the user's behalf on the grid.

Microsoft .Net Passport is the most widely deployed SSO service. Through a global user account comprising user's PUID, profile and credential, .Net Passport enable users moving easily between participating sites without needing to remember a specific set of credential for each site. Its underlying techniques include transparent http redirecting, SSL/TSL protocol, triple DES encryption and security cookies composed by ticket, profile and visited sites cookie [8].

Liberty Alliance project is going to establish a Federated Network Identity that links various user identities together. It would deliver the benefit of SSO to users by granting rapid access to resources to which they have permission, but it does not require the user's personal information to be stored centrally [9].

3 Extending the Semantics of Single Sign-On

Generally, SSO could be comprehended as an accumulation of a series authentications and authorizations, as shown in equation (1):

$$\text{Single Sign-On} = \sum (\text{authentication} + \text{authorization}) \quad (1)$$

However, authentication and authorization in such semantic are coupled too tightly to be separated from each other both in the program logics and the processing flows.

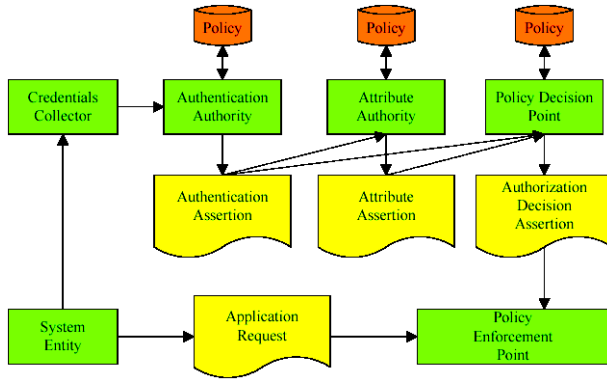


Fig. 1. This is the SAML authentication-authorization model. In this model, the result of authentication is authentication assertion, one kind of assertion that could be used to induce subsequent authorization decision assertion

According to OGSA security roadmap, future grid security services should leverage existing and emerging WS security specifications and XML standards as much as possible. As shown in figure 1, OASIS SAML is one of these choices [10].

SAML is an XML based framework for exchanging security information expressed in form of assertions about subjects. Assertions convey information about authentication actions that previously performed by three SAML authorities. Based on this architecture, SSO could be further comprehended as shown in equation (2):

$$\text{Single Sign-On} = \sum \text{authentication} + \sum \text{authorization} \quad (2)$$

Grid system intends to provide coordinated resources sharing, problem solving or services outsourcing in dynamic, multi-institutional virtual organizations, typical grid application often spread over multiple resource hosting sites and involve multiparty. For example, the computational power providers of certain distributed supercomputing application could be multiparty involved in a special computation job, the idle resource providers of certain high throughput application could be multiparty in-

volved in a special cryptographic problem, the search service providers of certain aggregate search engine could be multiparty involved in a special parallel searching, thousands of players participated in certain networking game could be multiparty involved in a special online competition, a set of web (grid) services constituting a workflow could be multiparty involved in a special services outsourcing plan, etc.

In these scenarios, it seems awkward for developers to regard SSO just as an accumulation of a series two-party authentication. Therefore, another new comprehension of SSO, as shown in equation (3), could be induced:

$$\begin{aligned} \text{Single Sign-On} = & \sum \text{two-party authentication} + \\ & \sum \text{multiparty authentication} + \sum \text{authorization} \end{aligned} \quad (3)$$

This comprehension brings some new semantics for SSO in grids:

1. Authentication and authorization actions of SSO could be designed as different grid security services, therefore, they might be implemented separately and communicate through requests/responses that convey SAML assertions.
2. Authentication service of SSO could support both two-party authentication and multiparty authentication that aims to confirm all principals asserted by different parties with satisfying confidence.

Multiparty relationships could be established in two main approaches: in the static approach, all participating parties must be known and presented in advance of authentication; in the dynamic approach, old participating parties might abandon the previous multiparty relationships while new participating parties might join the previous multiparty relationships. Obviously, the dynamic approach could be achieved by recursively using the static multiparty approach together with two-party approach. For convenience, this paper focuses on multiparty relationships that are formed in static approach unless explicitly stated.

4 Multiparty Joint Authentication

Denote each principal to be authenticated as a vertex, and let the edge connecting a pair of vertices represent that two principals have authenticated the counterparty mutually. A multiparty that involves n principals could be modeled as a graph of order n . We denote such a graph as MAG_n . After authenticating each pair of distinct principals, MAG_n would become a complete graph K_n with $n(n-1)/2$ edges.

A straightforward simplification is to choose one principal as a trusted third party, and let it to authenticate with the other principals mutually in turn. This simplification changes MAG_n into a complete bipartite graph $K_{1, n-1}$ called a star, needing only $n-1$ mutual authentications. A further simplification is to fully distribute the responsibility of the trusted third party and establish the MJA supposition as follows:

MJA Supposition. One principal could regard another principal as a trusted third party if either of the following conditions is satisfied:

1. Two principals have authenticated each other mutually.
2. Two principals have authenticated with one common trusted third party in advance.

Definition. Multiparty Joint Authentication (MJA) is to find a simplified or optimal authentication solution for a multiparty authentication in some security context that involves n principals, which is based on three conditions below:

1. If principal P_i , P_j were authenticated with one common trusted third party, then both P_i and P_j would confirm the counterparty with a specified, understood level of confidence even without real mutual authentication.
2. A principal and its trust third party must satisfy the MJA supposition.
3. There are m ($1 \leq m \leq n$) principals could act as a trusted third party to serve certain subsets comprising different principals. For short, let m be the order of the n member MJA, denote them as $n:m$.

Hamilton chain of the complete graph K_n is one possible MJA solution, however, it would not be the optimal answer if we take some practical restrictions into account:

1. Different authentication could have different QoS.
2. Users may insist performing mutual authentications for certain pairs of principals.
3. The MJA service provider may cache some mutual authentications between principals and the trusted third parties for a period of time.
4. The policies for a principal to become a trusted third party are of great varieties.
5. A principal may trust a trusted third party with different policies and security level.

These restrictions indicate how to find an optimal MJA solution face lots of challenges. The discussion about these algorithms is beyond the topic of this paper.

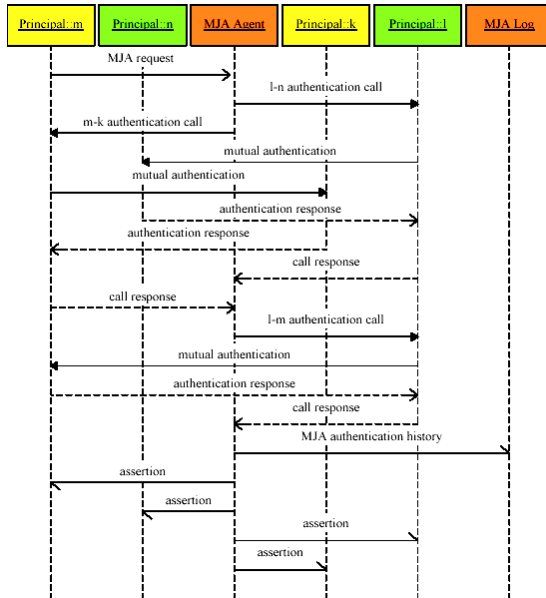


Fig. 2. This is a sequence diagram of a MJA scenario. It involves four principals, P_m , P_n , P_l , P_k , where the mutual authentications are performed by the principal pairs of P_l - P_n , P_m - P_k and P_l - P_m

5 Scenario of Multiparty Joint Authentication

A general scenario of MJA is shown in figure 2, which involves four principals, i.e., P_m , P_n , P_l , P_k , and the MJA service consists of two components, namely, MJA Agent and MJA Log. The sequence diagram of this scenario is captured as follows:

1. P_m sends its MJA request to the MJA Agent. The information of all involved principals and authentication polices are also wrapped in this request.
2. MJA Agent parses the request, finds a simplified MJA solution, and tries to invoke a series mutual authentication in accordance with the MJA solution. In this scenario, supposing the MJA should be conducted by the principal pairs of P_l - P_n , P_m - P_k and P_l - P_m . These mutual authentications could be invoked in a parallel way.
3. After standard mutual authentication, the principal invoked by MJA Agent should acknowledge to the invoker.
4. If all mutual authentications of MJA solution had been invoked and acknowledged, MJA Agent would issue a MJA assertion and a MJA history item with digital sign.
5. MJA history item should be sent to MJA Log for future auditing.
6. A MJA assertion could be cached, stored up by MJA Agent for future use, or, it may be directly distributed to each principal for subsequent activities.

An assertion is an XML message that may comprise 'major version', 'minor version', 'assertion identifier', 'issuer', 'issuer time', 'conditions', 'advice', 'digital signature', and one or more 'statement'. The XML Schema of assertion could be found in SAML specifications [10]. One assertion example is shown as follows:

```
<Assertion
AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
IssueInstant="2004-05-09T00:46:02Z"
Issuer="grid.sjtu.edu.cn"
MajorVersion="1"
MinorVersion="1"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Conditions
    NotBefore="2004-05-09T00:46:02Z"
    NotOnOrAfter="2004-05-09T00:51:02Z">
    <AudienceRestrictionCondition>
      <Audience>http://grid.sjtu.edu.cn/test#p_m</Audience>
      .....
    </AudienceRestrictionCondition>
  </Conditions>
  <AuthenticationStatement
    AuthenticationInstant="2004-05-09T00:46:00Z"
    AuthenticationMethod="http://grid.sjtu.edu.cn/mja">
    <Subject>
      <NameIdentifier>"Principal_m"</NameIdentifier>
      <SubjectConfirmation>
        <SubjectConfirmationData>
          <ds:X509Certificate>.....</ds:X509Certificate>
        </SubjectConfirmationData>
      </SubjectConfirmation>
    </subject>
    .....
  </AuthenticationStatement>
</Assertion>
```

```

<subject>.....</subject>
</AuthenticationStatement>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    .....
    <ds:SignedInfo>
      <ds:SignatureValue>.....</ds:SignatureValue>
      <ds:KeyInfo>
        .....
        </ds:KeyInfo>
      </ds:Signature>
    </Assertion>

```

6 OGSA Security Model and Multiparty Joint Authentication

MJA service should be regarded as a special service of authentication service for grids. As shown in figure 3, smooth interactions between MJA service and other grid security services would form a new SSO architecture for grid computing.

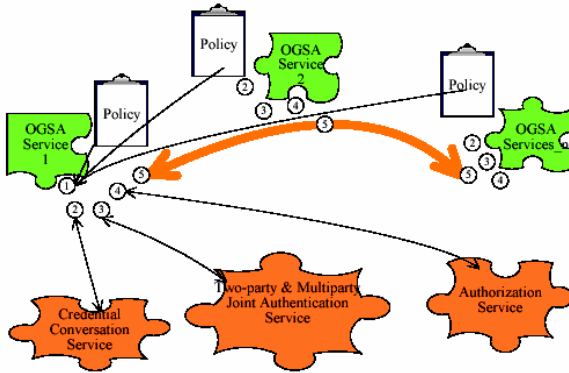


Fig. 3. The MJA service would interact smoothly with other grid security services to establish a security grid environment. Other security services, such as privacy service, SSL/TSL service, etc, may also be deployed in this new SSO architecture

As shown in figure 3, OGSA Service₁ wants to work with OGSA Service₂, ..., and OGSA Service_n together to accomplish a grid computing job. It could be achieved securely by following steps:

1. OGSA Service₁ sends its request to all services involved and collects published policies for each services.
2. Each service determines what security mechanisms and credentials are to be used. If the required credentials were not available, OGSA service would contact Credential Conversion Service to convert existing credentials to the needed format.
3. All services use the Authentication Service, including both two-party authentication service and multiparty joint authentication service, to authenticate some necessary principals and acquire a MJA assertion.

4. The MJA assertion is presented to Authorization Service and authorization decision assertions are produced for each principal.
5. If authorization were success, all services would be linked together to perform the grid computing job expected.

7 Conclusion

Single Sign-On is one of the most important requirements for grid system, which could be comprehended as an accumulation of a series of two-party authentication, multiparty authentication and authorization.

MJA is to find some simplified or optimal authentication solutions for multiparty authentication. This paper presents a formal definition for MJA, analyses its practical restrictions, illustrates MJA scenario with its internal sequence diagram and assertion format, and indicate that MJA service could be naturally regarded as a special grid security service based on OGSA security model.

References

1. Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., Takagi, M., Bing-Wo, R., Amir, A., Murakawa, R., Hernandez, O., Magowan, J., Bieberstein, N.: Introduction to Grid Computing with Globus. IBM Corp. (2002)
2. Surridge, M., Upstill, C.: Grid Security: Lessons for Peer-to-Peer Systems. In: Shahmehri, N., Graham, R., Garroni, R. (eds.): Proc. of the 3rd Intl. Conf. on Peer-to-Peer Computing. IEEE Press, Sweden (2003) 2-6
3. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. In: Gong, L., Reiter, M. (eds.): Proc. of the 5th ACM Conf. on Computer and Comm. Sec. ACM Press, New York (1998) 83-92
4. Volchkov, A.: Revisiting Single Sign-On: A Pragmatic Approach in a New Context. IT Pro. 1 (2001) 39-45
5. Dae-Hee, S., Im-Yeong, L., Soo-Young, C., Choon-Soo, K.: Single Sign-On Systems Using Multi-Agent System. In: Gebali, F. (ed.): Proc. of IEEE Pacific Rim Conf. on Comm., Computers and Sig. Processing. IEEE Press, Canada (2003) 692-695
6. Novotny, J., Tuecke, S., Welch, V.: An Online Credential Repository for the Grid: MyProxy. In: Williams, A. (ed.): Proc. of 10th Intl. Symposium on H. Performance Distributed Computing. IEEE Press, California (2001) 104-111
7. Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L., Tuecke, S.: Security for Grid Services. In: Azada, D. (ed.): Proc. of 12th Intl. Symposium on H. Performance Distributed Computing. IEEE Press, Washington (2003) 48-57
8. Opplinger, R.: Microsoft .Net Passport: A Security Analysis. Computer 4 (2003) 29-35
9. Website: <http://www.projectliberty.org/>
10. Website: <http://www.oasis-open.org/specs/index.php#samlv1.1>

A Software Engineering Perspective for Services Security

Jun Han

School of Information Technology
Swinburne University of Technology
John Street, Hawthorn, Vic. 3122, Australia
jhan@it.swin.edu.au

Abstract. Services are usually developed and deployed independently; and systems can be formed by composing relevant services to achieve set goals. In such an open and dynamic environment, security is of paramount importance. We have seen much work in the traditional area of information and network security, focusing on developing various security techniques. More recently, there have been efforts in integrating the security techniques into languages and infrastructural support that are used for developing services and systems. In fact, the development of services and the composition of service-based systems are software engineering activities. As such, they need to be viewed from a software engineering perspective. In this paper, we introduce an approach to services security engineering, to answer the questions like what the security properties of services and service-based systems are and how they meet the user's security requirements. It deals with the issues of (1) security property characterisation for services, (2) compositional security analysis for service-based systems, and (3) certification of services.

1 Introduction

Service oriented computing has promised a new way to deliver information technology support for individuals and businesses. Services in this context, including Web and Grid services, are software applications that are deployed over standard computing platforms, and are aimed at being integrated or composed with each other to form Internet-based systems and perform cross-application transactions. As the Internet is a hostile environment, security for services and their compositions is of great concern.

In the broad context of dealing with software and system security, the current practices adopt a *defensive*, *retrospective* and *reactive* line of thinking. That is, they tend to follow the path of patching up security holes found in systems that are often built without systematic security considerations [3], or security being treated as a “after-thought” or “add-on” in system development such as firewalls, sandboxes and security wrappers [12, 11]. While these may be the most practical ways available to deal with system security, it definitely does not represent a satisfactory situation. Instead, a more appropriate engineering approach should be taken. In general, this requires

- the understanding of the security risks and requirements for a system,
- the availability of security techniques that can be used, and
- the way of how the system can be developed using the security techniques to meet its security requirements.

While the development of various security techniques such as encryption algorithms and key exchange protocols has been the main topic of the information security community, there has been limited study into the software engineering aspects of system security, i.e., how to use the security techniques in software and system development to satisfy system security requirements in a *proactive* and *predictive* manner.

In this paper, we focus on the software engineering issues concerning the security of services and service compositions. In particular, we consider the following two aspects:

1. For an individual service, its specific (ensured and required) security properties need to be characterised and published so that the potential users can assess its suitability in given contexts of use.
2. For a service composition, we need to analyse the compatibility of the interacting services' (ensured and required) security properties and deduce the system-level security properties based on those of the individual services.

Only by achieving these two objectives can we answer the questions concerning the security of services and service-based systems. Otherwise, there will always be security concerns about composing independent third-party services, which will undermine the future of service oriented computing as a whole. In the remainder of this paper, we introduce an approach to services security engineering, aiming to address the above issues.

2 An Approach to Services Security Engineering

The security properties of a service will be part of and impact on the security of a system that uses it. As such, they must be characterised and made explicit so that the users can be aware of its security characteristics and use it with confidence. Another equally important aspect that impacts on the target system's security is the system's composition architecture that connects the services in a specific manner. In addressing the issue of security characterisation for services and service-based systems, our approach has three major components:

1. characterise and publish the security properties of individual services through the use, adaptation and formalisation of the Common Criteria [2],
2. certify the security properties of services against their implementations, and
3. analyse and deduce the security properties of a composed system in terms of the characteristics of its services and its composition architecture.

While our research has been mainly on the first and third aspects (characterisation and compositional analysis), we also give an account of the specific requirements for the second aspect (certification). Note that our approach is set in the context of a general framework for component-based software [6].

2.1 Characterisation and Publication of Service Security

The ISO/IEC International Standard 15408, *Common Criteria for Information Technology Security Evaluation, version 2.1* – commonly referenced as the *Common Criteria* or simply *CC* [2], identifies the various security requirements for IT products and systems, and provides a good starting point for characterising the security properties of services, i.e., with the services being regarded as IT products/systems. Using the Common Criteria, we are able to analyse and identify the types of security properties and the levels of security strength that a service has implemented. For example, a set of properties based on the Common Criteria can be used to characterise how user data is protected with which levels of strength.

As the Common Criteria are written in natural language and are not amenable to formal analysis. To provide precise and automated support for security analysis, the specification of security properties should take a more formal and succinct form than a lengthy informal document. We have analysed the security functional requirements of the Common Criteria and formulated a formal model for service security characterisation and specification. For a given service, we distinguish its ensured and required properties. A *required* property is one that has to be satisfied by a user (i.e., another service) when the user wants to use the service or a particular functionality of the service. An *ensured* property is one that the service guarantees to its users when performing certain functionality, which may be subject to certain required properties being met. A required or ensured security property is a formalised statement that states a fact about or dependency between certain security properties. It adopts a logic programming style. The following example illustrates our approach to security characterisation and specification for services.

Let us consider an online tax return processing system. The tax office provides an online tax-processing service that processes people's tax returns. People use a submission client (i.e., another service) to submit their tax forms containing all the required information. The tax processing service requires that the tax form be encrypted with the tax-processing service's public key before submission for the purposes of confidentiality and integrity. Some of the security properties associated with the tax-processing service t 's tax return lodgment functionality are as follows:

$$\begin{aligned} & \textit{owned}(k, \textit{this}). \\ & \textit{owned}(k^{-1}, \textit{this}). \\ & \textit{signed}(\textit{tax\_statement}, k^{-1}) \leftarrow \textit{encrypted}(\textit{tax\_form}, k), \textit{owned}(k, \textit{this}). \end{aligned}$$

The first two formulas state the properties that the service owns a public key k and private key k^{-1} . The third formula states that if the submitted tax form is encrypted with the tax processing service's public key, then the tax processing statement will be returned signed with the service's private key for verifying authenticity. Note that in this formula, there are two required property statements (in the tail of the formula) and one ensured property (i.e., the head of the formula). A further property could state that the tax statement is also encrypted using the submitter's public key.

On the other hand, the submission client c may have the following properties:

$$\begin{aligned} & \textit{owned}(k, t). \\ & \textit{encrypted}(\textit{tax\_form}, k). \\ & \textit{sees\_signed}(\textit{this}, \textit{tax\_statement}) \leftarrow \\ & \quad \textit{signed}(\textit{tax\_statement}, k^{-1}), \textit{owned}(k^{-1}, t). \end{aligned}$$

The first two formulas state that the submission client ensures that the tax form is encrypted with the tax-processing service's public key. The third formula states that the submission client requires that the tax processing statement be signed with the tax-processing service's private key so that it can verify the statement's authenticity by applying the tax-processing service's public key.

In general, the required and ensured security properties of a service together with their dependencies need to be included in a service's published description. In this way, the potential users can assess the service's suitability. In the above example, both the tax-processing service and the submission client can access each other's security properties and assess if the other service satisfies their own security requirements (see section 2.3 for further discussion).

2.2 Certification of Service Security

A service has an implementation and a description. In particular, the security property description of a service should reflect the security measures adopted in the service's implementation. In deploying such a service, however, how can a potential user be assured that the implementation actually conforms to the description and any unauthorised modification of the service (either the implementation or the description), be it accidental or malicious, can be easily detected? This is about the *integrity* of the service, and concerns both the service implementation and description. This issue is especially important in the context of dynamically configurable service-based systems, such as those using Web services, Grid services or mobile agents.

To satisfy the above integrity requirement for a service, the service description should be packaged together with the service implementation through techniques like introspection. Then the packaged service needs to be verified, certified, digitally stamped [4], and sealed by a certification authority. The certified service's description also contains certification-related information, including details about the certificate, certification stamp, validity period and so on, which can be revealed when queried. This information is read-only to other services, and can only be altered by the issuing certification authority.

In general, services can only be tested and certified individually, not within the context of the complete composed system [7]. The certification process involves the following tasks. First, the conformance between the service implementation and the service description needs to be checked, including the security properties. Second, the service description is approved and certified based on the result of the conformance check. Third, the service implementation and description is sealed for integrity. Finally, the issued certificate is registered so that

its authenticity can be verified by interested parties. The certificate contains a unique service identifier that is accessible from the service description by others.

The certified assurances must be verifiable statically and dynamically by other services or system integrators. Once certified, a re-compilation of the service would automatically erase all the relevant certification and identity related information. In fact, a tampered description or implementation would result in a void certificate, and this could be established by the contacting services from the information packaged with the service. If the service needs to alter its security properties, it requires a new certificate after the re-compilation. In general, such a certification scheme for services (including their implementations and descriptions) will significantly increase the user confidence in these services.

While the evaluation and certification of services is an important part of our approach, we primarily rely on existing technology and infrastructure as well as others' research in this regard. For example, the certification authority could be performed by government security evaluation agencies such as the Digital Signal Directorate in Australia. More on service or software component certification can be found in [4, 13].

2.3 Security Analysis for Service-Based Systems

A service-based system is a composition of individual services. These individual services are usually provided by third parties and consequently their development information is not available. To analyse the system's security characteristics, we have to rely on the published security properties of the individual services and its composition architecture. As such, we need a service-based, architecture-directed composition model for security analysis.

While highlighting the types of security properties for services and systems, the Common Criteria do not directly address system composition issues. In developing the security composition model, we need to consider the security compatibility of the services as dictated by the architectural interactions, the trade-offs and compromises between individual services' security strength in the system context, the derivation of system-wide properties from service properties and service interactions, the security impact of the overall architecture, and the relationships or dependencies between the system and its underlying enabling technologies (as part of the system's security environment). To date, we have focused on two of these issues, namely, the security compatibility between interacting services and the derivation/checking of system-wide security properties.

In the tax processing example given earlier, for instance, a question concerns whether or not the given tax-processing service and the tax submission client actually satisfy each other's security requirements for carrying out the tax lodgment activity. In fact, they do satisfy each other's requirements as follows: the submission client's ensured property of using the processing service's public key to encrypt its tax form satisfies the corresponding required property of the processing service; consequently, the processing service ensures that the tax statement will be digitally signed using its private key before sending to the submission client; in turn, this satisfies the submission client's corresponding

security requirement for receiving its tax statement. As such, the tax lodgment transaction can be carried out between the two services. On the other hand, let us assume that the submission client service also requires the tax statement being encrypted (for confidentiality) as well as signed. This additional requirement can not be met by the processing service. Consequently, the submission client will not (be able to) use that processing service.

Regarding the checking or derivation of system-wide security properties, let us extend the above example with a banking service. The banking service b provides a functionality for settling tax returns, i.e., receiving instructions from the tax processing service, transferring money between relevant accounts, and notifying both the tax processing service and the submission client about the tax settlement. For the interaction with the tax processing service, the banking service has the following security properties:

$$\begin{aligned} & \text{owned}(k2, \text{this}). \\ & \text{owned}(k2^{-1}, \text{this}). \\ & \text{signed}(\text{tax\_settlement}, k2^{-1}) \leftarrow \\ & \quad \text{encrypted}(\text{tax\_instruction}(c), k2), \text{owed}(k2, \text{this}). \end{aligned}$$

The relevant security properties of the tax processing service in relation to the banking service functionality are as follows:

$$\begin{aligned} & \text{owned}(k2, b). \\ & \text{encrypted}(\text{tax\_instruction}(c), k2). \\ & \text{sees\_signed}(\text{this}, \text{tax\_settlement}) \leftarrow \\ & \quad \text{signed}(\text{tax\_settlement}, k2^{-1}), \text{owned}(k2^{-1}, b). \end{aligned}$$

Note that the above two groups of formulas are very much similar to those concerning the relationship between the tax processing service and the submission client. We can see that the tax processing service t and the banking service b satisfy each other's security requirements, i.e., compatible at the peer level.

For the interaction with the submission client, the banking service has the following security properties:

$$\begin{aligned} & \text{signed}(\text{tax\_settlement}, k2^{-1}) \leftarrow \\ & \quad \text{encrypted}(\text{tax\_instruction}(c), k2), \text{owed}(k2, \text{this}). \end{aligned}$$

The above formula states that the tax settlement notice is also sent to the tax submission client. Correspondingly, the submission client has the following security properties:

$$\begin{aligned} & \text{sees\_signed}(\text{this}, \text{tax\_settlement}) \leftarrow \\ & \quad \text{signed}(\text{tax\_settlement}, k2^{-1}), \text{owned}(k2^{-1}, b). \end{aligned}$$

The formula states that the client sees its tax settlement from the banking service signed with the banking service's private key. In fact, this represents a system-wide property, which can only be deduced from combining the two compatible binary security compositions between the submission client c and the processing service t and between the processing service t and the banking service b .

We are currently developing a prototype tool kit that supports the publication of the security properties for services as part of their descriptions (just like their functional description information), and the compositional security reasoning for service-based systems, including analysis of peer-level security compatibility between services and derivation/checking of system-wide security properties.

3 Related Work

As mentioned earlier, there has been a long history and much work in developing techniques for information and network security. Encryption algorithms, digital signature schemes, security key exchange protocols and firewalls are just some of these techniques. They are the basic techniques for implementing system security, just like other programming techniques for implementing system functionality. We are all aware that basic programming techniques is not adequate for building large-scale complex systems. Similarly, basic security implementation techniques are not sufficient for dealing with the security of such complex systems.

Modularity has been an essential tool for dealing with software complexity, and has allowed us to introduce the concepts of software components and component-based systems [6]. Functionally, we need to describe *what* a component does, and to analyse how the functional requirements of the system are met based on those of the components. For security, we need the corresponding techniques for security description and compositional analysis [9]. As services are essentially independent software components, the same argument applies to services and service-based systems. Therefore, there is not only the need to have techniques for implementing security, but also the need for characterising and specifying the security properties of services and the need for analysing the security properties of service-based systems in meeting the systems' security requirements.

Following the development of component software, Web services and Grid services in recent years, there has been much effort in making security techniques standard constructs/libraries in various programming and service composition languages. They include Java security [5], Web Services Security [1] and other related standards such as WS-Trust, WS-SecureConversation and Security Assertion Markup Language (SAML). These efforts essentially provide implementation support for security. The issues of security property description and compositional analysis remain unaddressed.

Only in the past few years have we seen calls for moving security issues in services and systems to the next level, i.e., investigating the issues from a software engineering perspective [8]. It includes the characterisation of service/component security properties and the compositional security analysis for service/component-based systems [9, 10] as well as security certification [4]. Much more needs to be done to realise the goals of security-aware service-oriented computing with open dynamic service composition.

4 Conclusions

In this paper, we have introduced an approach to services security engineering. It includes security characterisation and description for services, and compositional security analysis for service-based systems. The approach and related techniques are set within a framework of service security certification. Our approach to security characterisation is partially based on the international security evaluation standard, the Common Criteria. The compositional analysis techniques allow us to check the security compatibility between interacting services and to verify whether or not the security requirements for a system are met.

Acknowledgment

I would like to thank Khaled Khan for his collaboration on security engineering for component software.

References

1. B. Atkinson et al. Web services security (WS-Security). Working Group Report <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>, IBM, April 2002.
2. Common Criteria Project/ISO. *Common Criteria for Information Technology Security Evaluation, version 2.1 (ISO/IEC International Standard 15408)*. NIST, USA and ISO, Switzerland, <http://csrc.nist.gov/cc/>, December 1999.
3. A. Ghosh, C. Howell, and J.A. Whittaker. Building software securely from the ground up. *IEEE Software*, 19(1):14–16, 2002.
4. A. Ghosh and G. McGraw. An approach for certifying security in software components. In *Proc. 21st National Information Systems Security Conference*, 1998.
5. L. Gong, G. Ellison, and M. Dageforde. *Inside Java 2 Platform Security*. Addison-Wesley, Reading, MA, USA, 2003.
6. J. Han. A comprehensive interface definition framework for software components. In *Proc. 1998 Asia-Pacific Software Engineering Conference*, pages 110–117, 1998.
7. J. Hopkins. Component primer. *Communications of the ACM*, 43(10):27–30, 2000.
8. IEEE. Special issue on building software securely. *IEEE Software*, 19(1), 2002.
9. K. Khan and J. Han. Security aware software composition. *IEEE Software*, 19(1):34–41, 2002.
10. K. Khan and J. Han. A security characterisation framework for trustworthy component based software systems. In *Proc. 27th Annual International Computer Software and Applications Conference*, pages 164–169, 2003.
11. P. Sewell and J. Vitek. Secure composition of insecure components. In *Proc. 12th IEEE Computer Security Foundations Workshop*, pages 136–150, 1999.
12. J. Voas. The challenges of using COTS software in component-based development. *IEEE Computer*, pages 44–45, 1998.
13. J. Voas. Certifying software for high-assurance environments. *IEEE Software*, (4):48–54, 1999.

Algorithms for Congestion Detection and Control*

Wu Liu^{1,2,3}, Hai-Xin Duan¹, Jian-Ping Wu¹, Xing Li¹, and Ping Ren^{2,4}

¹ Network Research Center of Tsinghua University, 100084 Beijing, China
{liuwu@cernet.edu.cn}

² Sichuan Agricultural University, 611830, Dujiangyan Sichuan, China

³ Computer Software Lab. of Sichuan Normal University, 611168 Chengdu, China

⁴ Business Management School of Sichuan University, 610064 Chengdu, China

Abstract. This paper proposes Early Congestion Detection and Control (ECDC) gateways for congestion avoidance which detects early congestion by computing the average queue size and can notify connections of congestion either by dropping packets or by setting a bit in packet headers. When the average queue size exceeds a preset threshold, the gateway drops or marks each arriving packet with a certain probability which is a function of the average queue size. ECDC gateways keep the average queue size low while allowing occasional bursts of packets in the queue. During congestion, the probability that the gateway notifies a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth through the gateway.

1 Introduction and Related Works

There are a number of mechanisms that have been proposed for transport-layer protocols to maintain high throughput and low delay in the network. Some of these proposed mechanisms are designed to work with current gateways [4][8], while other mechanisms are coupled with gateway scheduling algorithms that require per-connection state in the gateway [7]. In the absence of explicit feedback from the gateway, transport-layer protocols could infer congestion from the estimated bottleneck service time, from changes in throughput, from changes in end-to-end delay, as well as from packet drops or other methods. Nevertheless, the view of an individual connection is limited by the timescales of the connection, the traffic pattern of the connection, the lack of knowledge of the number of congested gateways, the possibilities of routing changes, as well as by other difficulties in distinguishing propagation delay from persistent queuing delay.

The method of monitoring the average queue size at the gateway, and of notifying connections of early congestion, is based on the assumption that it will continue to be useful to have queues at the gateway where traffic from a number of connections is multiplexed together, with FIFO scheduling. Not only is FIFO scheduling useful for

* This work is supported by grants from the National Natural Science Foundation of China (Grant No. #90104002 & #2003AA142080) and by Computer Software Lab. of Sichuan Normal University.

sharing delay among connections, reducing delay for a particular connection during its periods of burstiness [1], but also it scales well and is easy to implement efficiently. In an alternative approach, some congestion control mechanisms that use variants of Fair Queuing [5] or hop-by-hop flow control schemes [7] propose that the gateway scheduling algorithm make use of per-connection state for every active connection.

Several researchers have studied Early Random Drop (ERD) gateways as a method for providing congestion avoidance at the gateway.

Hashem [2] discusses some of the shortcomings of Random Drop2 and Drop Tail (DT) gateways, and briefly investigates ERD gateways. In the implementation of ERD gateways in [2], if the queue length exceeds a certain drop level, then the gateway drops each packet arriving at the gateway with a fixed drop probability. This is discussed as a rough initial implementation.

The Gateway Congestion Control Survey [6] considers the versions of ERD described above. The survey cites the results in which the ERD gateway is unsuccessful in controlling misbehaving users [10]. ERD gateways are not expected to solve all of the problems of unequal throughput given connections with different roundtrip times and multiple congested gateways. In [6], the goals of ERD gateways for congestion avoidance are described as “uniform, dynamic treatment of users (streams/flows), of low overhead, and of good scaling characteristics in large and loaded networks”. It is left as an open question whether or not these goals can be achieved.

This paper proposes a different congestion avoidance mechanism at the gateway, ECDC (Early Congestion Detection and Control) gateways, with somewhat different methods for detecting congestion and for choosing which connections to notify of this congestion.

2 The General ECDC Algorithm

The ECDC gateway calculates the average queue size, using a low-pass filter with an exponential weighted moving average, by comparing to two thresholds: a minimum threshold and a maximum threshold. When the average queue size is less than the minimum threshold, no packets are marked, while the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are in fact dropped, or if all source nodes are cooperative, this ensures that the average queue size does not significantly exceed the maximum threshold.

When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability P_a which is a function of the average queue size avg . Whenever a packet is marked, the probability that a packet is marked from a particular connection is roughly proportional to that connection's share of the bandwidth at the gateway.

Thus the ECDC gateway algorithm has two parts, one for computing the average queue size determines the degree of burstiness that will be allowed in the gateway queue, and another for calculating the packet-marking probability determines how frequently the gateway marks packets, given the current level of congestion. The goal

is for the gateway to mark packets at fairly evenly-spaced intervals, in order to avoid biases and to avoid global synchronization, and to mark packets sufficiently and frequently to control the average queue size.

Figure 1 shows a general algorithm for ECDC gateways, and section 4 discusses an efficient implementation of these algorithms.

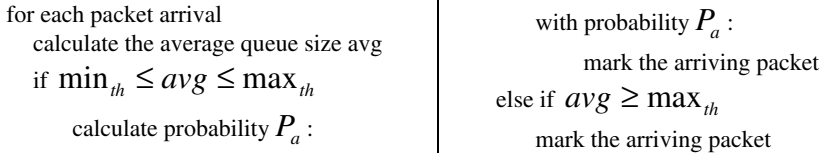


Fig. 1. General algorithm for ECDC gateways

The gateway's calculations of the average queue size take into account the period when the queue is empty (the idle period) by estimating the number m of small packets that could have been transmitted by the gateway during the idle period. After the idle period the gateway computes the average queue size as if m packets had arrived to an empty queue during that period.

As avg varies from $\min_{th}$ to $\max_{th}$, the packet-marking probability P_b varies linearly from 0 to $\max_p$: $P_b \leftarrow \max_p \cdot (avg - \min_{th}) / (\max_{th} - \min_{th})$

The final packet-marking probability P_a increases slowly as the count increases since the last marked packet: $P_a \leftarrow P_b / (1 - count \cdot P_b)$. This ensures that the gateway does not wait too long before marking a packet.

The gateway marks each packet that arrives at the gateway when the average queue size avg exceeds $\max_{th}$.

One option for the ECDC gateway is to measure the queue in bytes rather than in packets. With this option, the average queue size accurately reflects the average delay at the gateway. When this option is used, the algorithm would be modified to ensure that the probability that a packet is marked is proportional to the packet size in bytes:

$$P_b \leftarrow \max_p \cdot (avg - \min_{th}) / (\max_{th} - \min_{th})$$

$$P_b \leftarrow P_b \cdot \text{PacketSize} / \text{MaximumPacketSize}$$

$$P_a \leftarrow P_b / (1 - count \cdot P_b)$$

In this case, a large FTP packet is more likely to be marked than is a small TELNET packet. Section 3 discusses in detail the setting of the various parameters for ECDC gateways.

3 Calculations of Relative Parameters in ECDC

The low-pass filter is an exponential weighted moving average:

$$avg \leftarrow (1 - w_q) \cdot avg + w_q \cdot q$$

The weight w_q determines the time constant of the low-pass filter. This section discusses upper and lower bounds for setting w_q .

3.1 An Upper Bound for w_q

Theorem 1 $\sum_{i=1}^L ix^i = \frac{x + (Lx - L - 1)x^{L+1}}{(1-x)^2}$

Proof: Let $f(x) = \sum_{i=1}^L ix^i$, then $f(x)/x = \sum_{i=1}^L (ix^i/x) = \sum_{i=1}^L ix^{i-1}$,

So $\int f(x)/x dx = \int \sum_{i=1}^L ix^{i-1} dx = \sum_{i=1}^L \int ix^{i-1} dx = \sum_{i=1}^L x^i = \frac{x(1-x^L)}{1-x}$

Therefore $f(x)/x = \left(\frac{x(1-x^L)}{1-x} \right)' = \frac{1 + (Lx - L - 1)x^L}{(1-x)^2}$

And $f(x) = x \cdot \frac{1 + (Lx - L - 1)x^L}{(1-x)^2} = \frac{x + (Lx - L - 1)x^{L+1}}{(1-x)^2}$ ■

Now, we calculate avg_L . Assume that the queue is initially empty, with an average queue size of zero, then the queue increases from 0 to L packets over L packet arrivals. After the L^{th} packet arrives at the gateway, the average queue size avg_L is:

$$\begin{aligned} avg_L &= \sum_{i=1}^L i \cdot w_q (1-w_q)^{L-i} = w_q (1-w_q)^L \sum_{i=1}^L i \left(\frac{1}{1-w_q} \right)^i \\ &= L + 1 + \frac{(1-w_q)^{L+1} - 1}{w_q} \end{aligned}$$

For example, for $w_q = 0.001$, after a queue increase from 0 to 100 packets, the average queue size avg_{100} is 4.88 packets.

Given a minimum threshold $\min_{th}$, and given that we wish to allow bursts of L packets arriving at the gateway, then w_q should be chosen to satisfy the inequation:

$$L + 1 + \frac{(1-w_q)^{L+1} - 1}{w_q} < \min_{th}, \text{ for } avg_L < \min_{th}.$$

3.2 A Lower Bound for w_q

ECDC gateways are designed to keep the calculated average queue size avg below a certain threshold. However, this serves little purpose if the calculated average avg is not a reasonable reflection of the current average queue size. If w_q is set too low, then avg responds too slowly to changes in the actual queue size. In this case, the gateway is unable to detect the initial stages of congestion.

Assume that the queue changes from empty to one packet, and that, as packets arrive and depart at the same rate, the queue remains at one packet. Further assume that initially the average queue size was zero. In this case it takes $-1/\ln(1-w_q)$ packets arrival (with the queue size remaining at one) until the average queue size avg reaches $1-1/e=0.63$ [9]. For $w_q = 0.001$, this takes 1000 packet arrivals; for $w_q = 0.002$, this takes 500 packet arrivals. In most of our simulations we use $w_q = 0.002$.

3.3 Setting $\min_{th}$ and $\max_{th}$

The optimal values for $\min_{th}$ and $\max_{th}$ depend on the desired average queue size. If the typical traffic is fairly bursty, then $\min_{th}$ must be correspondingly large to allow the link utilization to be maintained at an acceptably high level. For the typical traffic in our simulations, for connections with reasonably large delay-bandwidth products, a minimum threshold of one packet would result in unacceptably low link utilization. The discussion of the optimal average queue size for a particular traffic mix is left as a question for future research.

The optimal value for $\max_{th}$ depends in part on the maximum average delay that can be allowed by the gateway.

The ECDC gateway functions most effectively when $\max_{th} - \min_{th}$ is larger than the typical increase in the calculated average queue size in one roundtrip time. A useful rule-of-thumb is to set $\max_{th}$ to at least twice of $\min_{th}$.

3.4 Calculation of the Average Queue Length

The initial packet-marking probability P_b is calculated as a linear function of the average queue size in which the number of arriving packets between marked packets is a uniform random variable: $P_b \leftarrow \max_p (avg - \min_{th}) / (\max_{th} - \min_{th})$

The parameter $\max_p$ gives the maximum value for the packet-marking probability P_b , achieved when the average queue size reaches the maximum threshold.

The Uniform Random Variables. Let X be a uniform random variable from $\{1, 2, \dots, \left\lceil \frac{1}{P_b} \right\rceil\}$. This is achieved if the marking probability for each arriving packet is $P_b / (1 - \text{count} \cdot P_b)$, where *count* is the number of unmarked packets that have arrived since the last marked packet. In this case,

$$\Pr(X = n) = \frac{P_b}{1 - (n-1)P_b} \prod_{i=0}^{n-2} \left(1 - \frac{P_b}{1 - i \cdot P_b} \right) = P_b \text{ for } 1 \leq n \leq \left\lceil \frac{1}{P_b} \right\rceil,$$

$$\text{and } \Pr(X = n) = 0 \text{ for } n > 1/P_b, \quad E[X] = \frac{1}{2P_b} + \frac{1}{2}$$

In this paper, we set $\max_p = 1/50$. When the average queue size is halfway between $\min_{th}$ and $\max_{th}$, the gateway drops, on the average, roughly one out of 50 (or one out of $1/\max_p$) of the arriving packets. ECDC gateways perform best when the packet-marking probability changes fairly slowly as the average queue size changes; this helps to discourage oscillations in the average queue size and in the packet-marking probability.

4 Implementation of the Optimized ECDC Algorithm

For every packet arrival at the gateway queue, the ECDC gateway calculates the average queue size. This can be implemented as follows:

$$avg \leftarrow avg + w_q (q - avg)$$

As long as w_q is chosen as a (negative) power of two, this can be implemented with one shift and two additions (given scaled versions of the parameters) [3].

Because the ECDC gateway computes the average queue size at packet arrivals, rather than at fixed time intervals, the calculation of the average queue size is modified when a packet arrives at the gateway to an empty queue. After the packet arrives at the gateway to an empty queue the gateway calculates m , the number of packets that might have been transmitted by the gateway during the time that the line was free. The gateway calculates the average queue size as if m packets had arrived at the gateway with a queue size of zero. The calculation is as follows:

$$m \leftarrow (time - q\_time) / s$$

$$avg \leftarrow (1 - w_q)^m avg$$

Where q_time is the start of the queue idle time, and s is a typical transmission time for a small packet. This entire calculation is an approximation, as it is based on the number of packets that might have arrived at the gateway during a certain period of time. After the idle time $(time - q\_time)$ has been computed to a rough level of

accuracy, a table lookup could be used to get the term $(1 - w_q)^{(time - q\_time) / s}$, which could itself be an approximation by a power of two.

When a packet arrives at the gateway and the average queue size avg exceeds the threshold $\max_{th}$, the arriving packet is marked. There is no recalculation of the packet-marking probability. However, when a packet arrives at the gateway and the average queue size avg is between the two thresholds $\min_{th}$ and $\max_{th}$, the initial packet-marking probability P_b is calculated as follows:

$$P_b \leftarrow C_1 avg - C_2, \text{ Where } C_1 = \frac{\max_p}{\max_{th} - \min_{th}}, C_2 = \frac{\max_p \min_{th}}{\max_{th} - \min_{th}}$$

The parameters $\max_{th}$, $\min_{th}$, and $\max_p$ are fixed parameters that are determined in advance. The values for $\min_{th}$ and $\max_{th}$ are determined by the desired bounds on the average queue size, and might have limited flexibility. The fixed parameter $\max_p$, however, could easily be set to a range of values. In particular, $\max_p$ could be chosen so that C_1 is a power of two. Thus, the calculation of P_b can be accomplished with one shift and one add instruction.

It is possible to implement the ECDC gateway algorithm to use a new random number only once for every marked packet, instead of using a new random number for every packet that arrives at the gateway when $\min_{th} \leq avg < \max_{th}$. When the average queue size is constant the number of packet arrivals after a marked packet until the next packet is marked is a uniform random variable from $\{1, 2, \dots, \lceil 1/P_b \rceil\}$. Thus, if the average queue size was constant, then after each packet is marked the gateway could simply choose a value for the uniform random variable $R = \text{Random}[0, 1]$, and mark the n^{th} arriving packet if $n \geq R/P_b$. Because the average queue size changes over time, we re-compute R/P_b each time when P_b is recomputed. If P_b is approximated by a negative power of two, then this can be computed using a shift instruction instead of a divide instruction.

The following algorithm gives the pseudo-code for an efficient version of the ECDC gateway algorithm.

5 Conclusion

ECDC gateways are an effective mechanism for congestion avoidance at the gateway, in cooperation with network transport protocols. If ECDC gateways drop packets when the average queue size exceeds the maximum threshold, rather than simply setting a bit in packet headers, then ECDC gateways control the calculated average queue size. This action provides an upper bound on the average delay at the gateway.

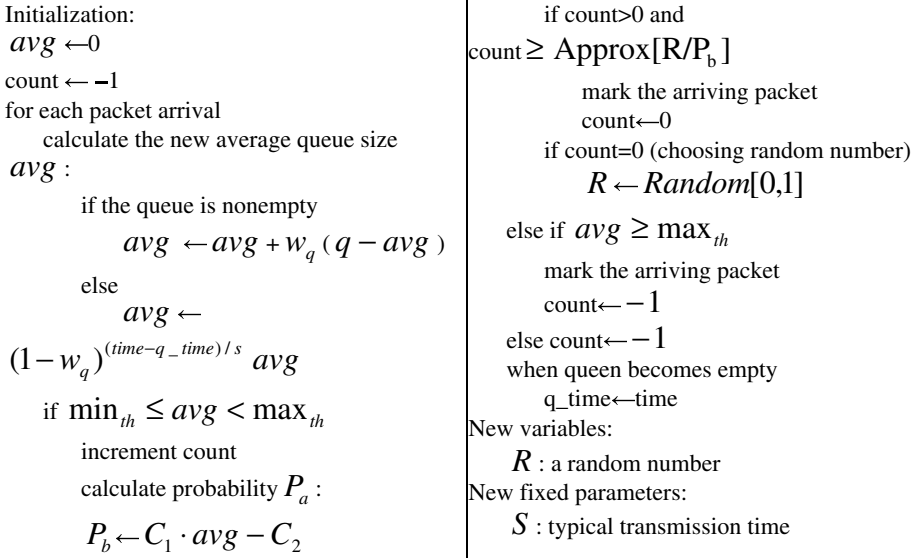


Fig. 2. Efficient algorithm for ECDC gateways

References

1. Clark, D.D., Shenker, S., and Zhang, L., "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", SIGCOMM '02, August 2002, p. 14-26.
2. Hashem, E., "Analysis of random drop for gateway congestion control", Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 2002.
3. Jacobson, V., "Congestion Avoidance and Control", Proceedings of SIGCOMM '98, August 1998, pp. 314-329.
4. Jain, R., "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks", Computer Communication Review, V.19 N.5, October 2002, pp. 56-71.
5. Keshav, S., "A Control-Theoretic Approach to Flow Control", Proceedings of SIGCOMM '91, September 2002, p.3-16.
6. Mankin, A. and Ramakrishnan, K. K., editors for the IETF Performance and Congestion Control Working Group, "Gateway congestion control survey", RFC 1254, August 1991.
7. Mishra, P., and Kanakia, H., "A Hop by Hop Rate-based Congestion Control Scheme", Proceedings of SIGCOMM '92, August 1992, p.112-123.
8. Mitra, D. and Seery, J., Dynamic Adaptive Windows for High Speed Data Networks: Theory and Simulations, Proceedings of SIGCOMM '00, September 2000, p.30-40.
9. Young, P., "Recursive Estimation and Time-Series Analysis", Springer-Verlag, 1984, pp. 60-65.
10. Zhang, L., "A New Architecture for Packet Switching Network Protocols", MIT/LCS/TR-455, Laboratory for Computer Science, Massachusetts Institute of Technology, August 1999.

Modeling Time-Related Trust

Chenlin Huang, Hua-Ping Hu, and Zhiying Wang

School of Computer, National University of Defense Technology
Changsha, 410073, Hunan, P.R. China
Fred_huang@tom.com, hphu@nudt.edu.cn

Abstract. Most of trust models today treat trust as a quantitative constant, focusing on the representation and the operations of uncertain believes, but ignore a fact that some believes are changing with time. To describe the change of trust relationship, the notion of Time-related Trust is introduced in this paper and the relationship between time and trust are discussed. Then time-related opinion is defined to represent time-related trust relationship based on Jøsang's work. A trust model based on it is also presented for modeling and reasoning about the time-related trust. With the model we further our analysis and discussion about the effects of time in trust and some properties of time-related opinion are concluded. Our work is helpful for understanding the dynamic property of trust and making the management of trust relationships more rational.

1 Introduction

Trust and trust management have been a hot topic in computer science recently due to the rapid development of Internet and e-business. Researches on trust relationship and trust model try to solve trust representation and operation in a quantitative way for the requirements of trust in e-business.

Many trust models have been proposed until now: trust model using direct and recommendation trust [1], the distributed trust model with a recommendation protocol by Alvarez et al [2], trust model based on Dempster-Shafer Theory [3], D. W. Manchala's trust model using fuzzy logic [4] and A. Jøsang's Subjective logic [5],[6],[7]. Most researches have shown the fact that trust relationship is changeable with time, but little has been done to show the detail relationship between trust and time. So far, trust models treat trust as a constant belief once its value is set. This paper focuses on the time-related trust relationship. Based on the analysis of the effects of time in trust, we extend Jøsang's subjective logic to a time-related trust model that attempts to address the concerns raised here.

2 The Effects of Time in Trust

Trust relationship is a binary relationship between trustor and trustee and is associated with certain properties, a specific context and the domain applied. This trust

relationship may dynamically change with time, trustor's experiences and the context. What we want to highlight here is the effects of time on trust relationship.

Firstly, we want to emphasize that not all kinds of trust relationships are influenced by time. Some trust relationships are naturally not influenced by time, once it is established it keeps its trust level until the appearance of new evidence. While, other trust relationships as binary trust (also called absolute trust) can only be either TRUE or FALSE, and are not much related with time. To better understand this, we define the trust relationship with one of the following features as *time-related trust relationship*:

1. The quantified belief by a trustor may change with time;
2. The trust relationship is associated with time;
3. The trust properties of a trustee are associated with time.

Here are examples for time-related trust relationships. Custom A had a belief that Company B is a good seller when A had some successful deals with B in 2001. While, A is not be so sure about B when A prepares to buy other things now. Another example: A's confidence in company B may decrease with time if A had expected that B was going down at the time trust was established, though no signs have ever been found by A. As to the case three, temperature and stock give us a good example.

Secondly, time is one of the key elements that make trust relationship a dynamic relationship. People have reached a consensus that trust is a dynamic concept [8-9]. The changes of a trust relationship are caused by certain new events such as: time, the change of trust context and new evidence. Their contributions to the change of trust are decided by trust relationship itself and differ from certain level to none. The change of trust context is out of consideration here because it usually causes the trust relationship to be a new one. New evidence that is studied in some current trust models in the form of experience [9], reputation [10] or probability event can be viewed as a time t event. So, we can build a general time-related trust model to deal with the new evidences. It can also handle the situation where new evidence has different weight as old one in some cases.

Last, the effects of time can be concluded here. It expresses a decrease of trustor's belief as time goes. The decrease is decided either by trustor's subjective opinion or by trustee's time-related properties. It expresses trustor's expectation or assumption to the trustee's behavior during the time without the appearance of new evidence. It reflects the change of time-related properties in a trust relationship. It reflects the different weight of time t belief and time s belief.

So, the study of time-related trust model will leads to a better understanding about the trust problems and a better trust model to be cored into applications.

3 Time-Related Trust Model

3.1 The Opinion Space

In order to find a simple intuitive representation of uncertain probabilities for time-related trust, we extend the definition of *opinion* in [11] to time-related opinion with

four functions to describe the change of opinion with time. Below is a full representation of time-related opinion.

Definition 1 (Time-Related Opinion) Let $\tilde{\Theta}^{-1}$ be a binary frame of discernment with 2 atomic states x and $\neg x$, and let $m_{\tilde{\Theta}}$ be a BMA [12] on $\tilde{\Theta}$ where $b_{x,t}$, $d_{x,t}$, $u_{x,t}$ and $a_{x,t}$ represent the belief, disbelief, uncertainty and relative atomicity functions on x at time t in $2^{\tilde{\Theta}}$ respectively. And use functions $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$ to represent the expected change of $b_{x,t}$, $d_{x,t}$, $u_{x,t}$ and $E_{x,t}$ within time Δt respectively. Then the opinion about x at time t , denoted by $w_{x,t}$, is the tuple defined by:

$$w_{x,t} \equiv (b_{x,t}, d_{x,t}, u_{x,t}, a_{x,t}, f(\Delta t), g(\Delta t), h(\Delta t), E(\Delta t)), \Delta t \geq 0, \quad (1)$$

$$f(0) = g(0) = h(0) = E(0) = 0$$

And the same opinion at time s , denoted by $w_{x,s}$, where $s \geq t$ can be computed with:

$$\begin{aligned} b_{x,s} &= b_{x,t} + f(s-t) \\ d_{x,s} &= d_{x,t} + g(s-t) \\ u_{x,s} &= u_{x,t} + h(s-t) \\ E_{w_{x,s}} &= E_{w_{x,t}} + E(s-t) \\ a_{x,s} &= \frac{[E_{x,s} - b_{x,s}]}{u_{x,s}}. \end{aligned} \quad (2)$$

For compactness, we also define a **T function** to express Eq. 2 as $w_{x,s} = T(w_{x,t})$.

In fact, this definition is a combination of opinion by Jøsang and time-related functions. Jøsang's definition can be viewed as a special case of time-related opinion where time does not take effect. The three coordinates $(b_{x,t}, d_{x,t}, u_{x,t})$ represent one's belief about a proposition at time t .

Theorem 1 (Time-Related Belief Function Additivity)

At any time t , the opinion $w_{x,t}$ satisfies:

$$b_{x,t} + d_{x,t} + u_{x,t} = 1, \quad x \in 2^{\tilde{\Theta}}, \quad x \neq \Phi. \quad (3)$$

and the change of $w_{x,t}$ satisfies:

$$f(\Delta t) + g(\Delta t) + h(\Delta t) = 0, \quad \Delta t \geq 0. \quad (4)$$

¹ The symbol $\tilde{\Theta}$ denotes the focused frame of discernment defined by Jøsang in [11].

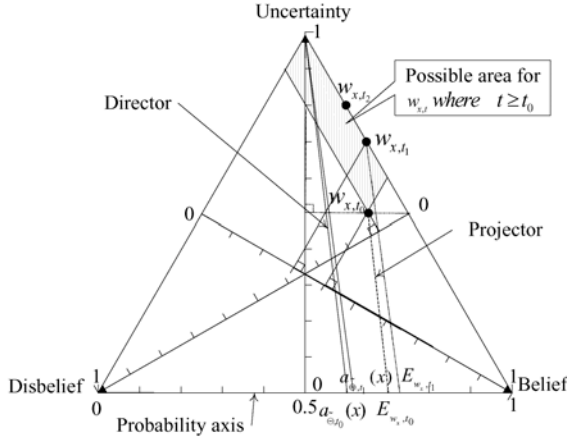


Fig. 1. As an example the position of opinion $w_{x,t_0} = (0.40, 0.10, 0.50, 0.60, f(\Delta t), g(\Delta t), h(\Delta t), E(\Delta t))$ is indicated as a point in the triangle. Also shown are the probability expectation value and the relative atomicity.

All $(b_{x,t}, d_{x,t}, u_{x,t})$ are dependent through Eq. 3 and time-related functions are dependent through Eq. 4. One of the functions $f(\Delta t)$, $g(\Delta t)$ and $h(\Delta t)$ is redundant, we still leave it in our definition for better understanding and convenience in practice.

The definition of opinion is in a binary frame of discernment which is in fact a focused frame of discernment [11]. As pointed out in [11], $a_{x,t}$ is a constructed value, which represents the weighted average of relative atomicities of x to all other states in 2^{Θ} and to keep the expected probability unchanged. So it is necessary for us to define $E(\Delta t)$ to make the calculation of $a_{x,t}$ possible with Eq. 4 in [11].

3.2 Further Discussion of Time-Related Opinion

Eq. 3 defines a triangle that can be used to graphically illustrate opinions as shown in Fig.1.

The definition of time-related opinion highlights the dynamic property of time-related trust relationship with the functions $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$. According to Eq. 2, the change of w_{x,t_1} can be rewritten as:

$$\begin{aligned} \Delta b_{x,t} &= b_{x,t_1} - b_{x,t_0} = f(t_1 - t_0) = f(\Delta t) \\ \Delta d_{x,t} &= d_{x,t_1} - d_{x,t_0} = g(t_1 - t_0) = g(\Delta t) \\ \Delta u_{x,t} &= u_{x,t_1} - u_{x,t_0} = h(t_1 - t_0) = h(\Delta t) \\ \Delta E_{w_{x,t}} &= E_{w_{x,t_1}} - E_{w_{x,t_0}} = E(t_1 - t_0) = E(\Delta t) \end{aligned} \quad (5)$$

² We use the symbol Θ to denote the corresponding frame of discernment in this paper.

Eq. 5 reflects the changes of one's belief as time goes. The change of opinion can be explained with the belief model in [11] as the reassignments of BMAs in the frame of discernment Θ [12]. According to Def. 2, 3, 4 in [11] the value of $b_{x,t}$, $d_{x,t}$, and $u_{x,t}$ will change respectively. While the value of $a_{\Theta}(x)$ will not change as Def. 5 in [11] defines.

The reassignments of BMAs cause the change of opinion $w_{x,t}$. It reflects the trustor's expectation about the coming evidence, the change of opinion or the change of trust relationship. Then, $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$ can be viewed as the degree of such a change in Δt . Ideally, $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$ should be the objective descriptions of the $\Delta b_{x,t}$, $\Delta d_{x,t}$, $\Delta u_{x,t}$ and $\Delta E_{w_{x,t}}$ respectively. However, it is not possible to find out the distribution function of one's belief just based on the evidences during a period of observation. Basically, those functions are the mixed results of objective evidences and subjective opinions. So, the time-related opinion in fact is a possible opinion at certain time with uncertainty, which is limited in the dotted area in Fig. 1. On the other hand, it reveals the fact that the key point in time-related trust computing is the construct of $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$.

An assumption must be emphasized that no new evidence appears during the time Δt in time-related opinion. Because when new evidence appears, both the trust value and the belief functions may change. The new opinion formed is more reliable than the old one and should be taken as the new computing point.

The reassignments of BMAs in Δt will influence opinion both in Θ and $\tilde{\Theta}$, as showed in Fig. 2. The procedure of reassignment is not random. Since there is no new evidence in Δt , it is reasonable to believe that:

1. For $b(x) = \sum_{y \subseteq x} m_{\Theta}(y)$, $m_{\Theta}(y)$ will not increase;
2. For $d(x) = \sum_{y \cap x = \Phi} m_{\Theta}(y)$, $m_{\Theta}(y)$ will not increase;
3. For $u(x) = \sum_{y \cap x \neq \Phi \wedge y \not\subseteq x} m_{\Theta}(y)$, $m_{\Theta}(y)$ will not decrease;

This can be viewed as the increase of one's doubts with time. An agent will not be so sure that one of the states in $y \subseteq x$ is true and the states in $y \not\subseteq x$ are false after time Δt . Also, the agent's belief masses on uncertain states will increase, that is to say the BMA will not decrease for every state in $y \not\subseteq x \wedge y \cap x \neq \Phi$. The result reflects the fade of one's belief and disbelief with time. Then for $w_{x,t}$ in Δt , $b_{x,t}$ and $d_{x,t}$ will not increase while $u_{x,t}$ will not decrease for Eq. 2. We express this as theorem 2:

Theorem 2: *The increase of uncertainty function is equal to the sum of the decrease of belief function and the disbelief function in a time-related opinion, i.e. $h(\Delta t) = -(f(\Delta t) + g(\Delta t))$.*

The properties of $f(\Delta t)$, $g(\Delta t)$ and $h(\Delta t)$ are decided by multi-effects and can either be continuous or discrete. It is necessary to decide every function for each time-related trust relationship. But the basic forms of $f(\Delta t)$, $g(\Delta t)$, $h(\Delta t)$ and $E(\Delta t)$ can still be given below based on our discussion so far:

$$\begin{aligned} f(\Delta t) &= \begin{cases} 0 & \Delta t = 0 \\ f(\Delta t) \leq 0 & \Delta t \in (0, \Delta t_k) \\ 0 & \Delta t \geq \Delta t_k \wedge b_{x,t+\Delta t_k} = 0 \end{cases}; g(\Delta t) = \begin{cases} 0 & \Delta t = 0 \\ g(\Delta t) \leq 0 & \Delta t \in (0, \Delta t_i) \\ 0 & \Delta t \geq \Delta t_i \wedge d_{x,t+\Delta t_i} = 0 \end{cases} \\ h(\Delta t) &= \begin{cases} 0 & \Delta t = 0 \\ h(\Delta t) \geq 0 & \Delta t \in (0, \Delta t_m) \\ 0 & \Delta t \geq \Delta t_m \wedge u_{x,t+\Delta t_m} = 1 \end{cases}; E(\Delta t) = f(\Delta t) + h(\Delta t)k \quad k \in (0, 1)^3. \end{aligned} \quad (6)$$

As to the relative atomicity, it has to be considered separately in Θ and in $\tilde{\Theta}$ (Fig. 2). $a_{\Theta}(x)$ keeps unchanged in Θ because the states in the frame of discernment keep unchanged in Δt . The reassignment of BMA will not influence the value of $a_{\Theta}(x)$ according to the definition of relative atomicity. While, things become different in $\tilde{\Theta}$. In $\tilde{\Theta}$, $a_{x,t}$ is a constructed value in order to keep the expectation probability unchanged [11]. That make it difficult to decide the change of $a_{x,t}$ and $E(w_{x,t})$.

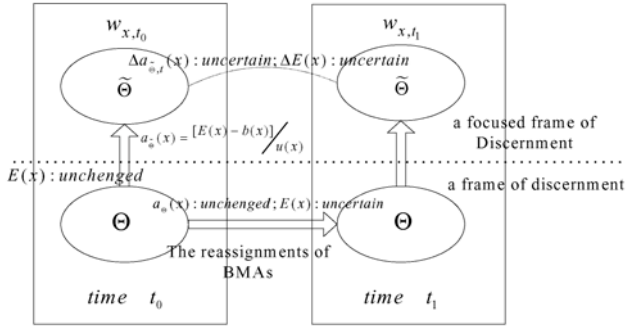


Fig. 2. The changes of $w_{x,t}$ at time t_1 and t_0 in Θ and $\tilde{\Theta}$.

To further explore the relationship between $a_{x,t}$, $E(w_{x,t})$ and Δt , we begin with the Def. 6 and Def. 8 in [11] and reach the conclusion that:

Theorem 3 (The relationship between relative atomic, expectation probability and time): *Given a frame of discernment Θ with a BMA m_{Θ} , and let $\tilde{\Theta}$ be the focused*

³ This formula can be induced based on Shafer's belief model [12] and Def. 6 and Def. 8 in [11].

frame of discernment with focus on x , and let $w_{x,t}$ be the corresponding time-related opinion, then except for the case when $\Delta d_{x,t} = 0 \wedge \Delta u_{x,t} \neq 0, \Delta E_{w_{x,t}} < 0$, the change of probability expectation function $E_{w_{x,t}}$ and the change of $a_{\Theta}(x)$ in Δt are uncertain.

The theorem 3 can be proven based on Shafer's belief model [12] and Jøsang's definitions in [11]. Now, we use an example to show the reassignments of BMAs.

3.3 Example: The Reassignments of BMAs

For example, the transition from the original frame of discernment with states $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ to a focused frame of discernment which focuses on the state $x_7 = (x_2 \cup x_3)$ is illustrated in Fig.3.

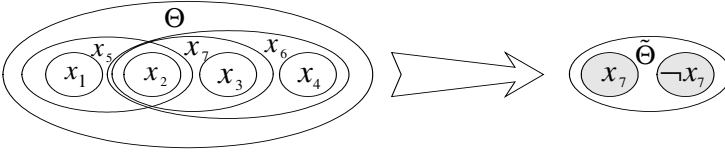


Fig. 3. Deriving the focused frame of discernment with focus on x_7 .

The assignments of BMAs and the time-related opinions at time t_0 , t_1 and t_2 are listed below in Table 1 and Table 2 respectively.

Table 1. The reassignments of BMAs for x_7 .

Time	$m_{\Theta}(x_1)$	$m_{\Theta}(x_2)$	$m_{\Theta}(x_3)$	$m_{\Theta}(x_4)$	$m_{\Theta}(x_5)$	$m_{\Theta}(x_6)$	$m_{\Theta}(\Theta)$
t_0	0.10	0.20	0.20	0.00	0.10	0.30	0.10
t_1	0.00	0.20	0.10	0.00	0.10	0.50	0.10
t_2	0.00	0.10	0.10	0.00	0.20	0.50	0.10

This produces the following time-related opinions for x_7 :

Table 2. The changes of time-related opinions for x_7 .

Time	$b(x_7)$	$d(x_7)$	$u(x_7)$	$E(x_7)$	$a_{\Theta}(x_7)$
t_0	0.40	0.10	0.50	0.70	0.60
t_1	0.30	0.00	0.70	0.73	0.62
t_2	0.20	0.00	0.80	0.64	0.55

It can be seen that both the change of $E(x)$ and the change of $a_{\Theta}(x)$ are uncertain.

4 Conclusion

Time plays an important role in describing the dynamic property of a trust relationship. In this paper, we present a trust model base on the notion of time-related opinion and have thoroughly discussed the effects of time in trust. Based on Jøsang's work on subjective logic, we have also defined the operators on time-related opinions, which can be used to handle the propagation and combination of trust, but due to the limitation of the paper's space, the extension parts are present in [13].

This model can be used to model the dynamic property of trust relationship in a quantitative way and we believe it is very general and can be successfully applied in a multitude of applications.

Our future work includes researching on the initialization of time-related opinion and a trust management system for distributed systems based on time-related trust model.

References

1. T. Beth, M. Borchertding, B. Klein: Valuation of trust in open networks. In Proceedings of the European Symposium on Research in Computer Security. Springer-Verlag, Brighton UK (1994) 3-18
2. A. Abdul-Rahman, S. Halles: A distributed trust model. In new Security Paradigms Workshop '97. (1997) 48-60
3. Y. Teng, V. Phoha, B. Choi: Design of trust metrics based on dempstershafer theory. <http://citeseer.nj.nec.com/461538.html>
4. D. W. Manchala: Trust metrics, models and protocols for electronic commerce transactions. In The 18th International Conference on Distributed Computing Systems. (1998) 3-12
5. A. Jøsang: Prospectives for modelling trust in information security. In Australasian Conference on Information Security and Privacy. (1997) 2-13
6. A. Jøsang: A subjective metric of authentication. In ESORICS: European Symposium on Research in Computer Security. Lecture Notes in Computer Science, Springer-Verlag, (1998)
7. A. Jøsang: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 9(3): (2001) 279-311
8. P. Lamsal: Understanding Trust and Security. <http://citeseer.nj.nec.com/lamsal01understanding.html> (2001)
9. Grandison T.: Trust Management for Internet Applications. PhD thesis. (2003)
10. Li Xiong, Ling Liu: A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities. The IEEE Conference on E-Commerce. (2003)
11. A. Jøsang: A Logic for Uncertain Probabilities. In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol.9, No.3(June 2001)
12. G. Shafer: A Mathematical Theory of Evidence. Princeton University Press. (1976)
13. Huang Chenlin, Hu huaping, Wang Zhiying: The Extension of Subjective Logic for Time-related Trust. Submitted to CTCIS'2004. (2004)

Defending DDoS Attacks Using Network Traffic Analysis and Probabilistic Packet Drop

Jungtaek Seo¹, Cheolho Lee¹, and Jongsub Moon²

¹ National Security Research Institute

62-1 Hwaam-dong, Yuseong-gu, Daejeon, 305-348, Republic of Korea
{seojt,chlee}@etri.re.kr

² CIST, KOREA University

1-Ga, Anam-dong, Sungbuk-Gu, Seoul, Republic of Korea
jsmoon@korea.ac.kr

Abstract. This research presents Traffic Rate Analysis (TRA) to efficiently analyze network traffic and a defense mechanism for DDoS attacks. TRA is defined as the ratio of a specific type of packets among the total amount of network packets, and divided into TCP flag rate and Protocol rate. By using the TRA for the network traffic, the normal and abnormal network traffic can be obviously distinguished from each other. Furthermore, to defense DDoS attacks, we probabilistically drop the network packets if their occurrence rates exceed the normal traffic rates. We expect that our proposed mechanism for analyzing network traffic and defending DDoS attacks will be very useful to early detect DDoS attacks and to protect TCP-based servers (e.g. Web servers) against DDoS attacks.

1 Introduction

Distributed Denial-of-Service (DDoS) attacks can temporarily disable the services provided by a target system or make harm to the system itself by exhausting network resources (e.g. memory, CPU, network bandwidth, etc) of the system with a huge amount of flooding traffic in a short time. As we can see in the incidents of DDoS attacks against commercial Web servers like *Yahoo*, *e-Bay*, and *E-Trade*, almost all the computer systems connected to the Internet are exposed to DDoS attacks [3,13].

Since DDoS attacks can have a harmful effect on all networked systems, they are regarded as a serious problem over the world. Many researches for detecting and defending the DDoS attacks are ongoing [4,5,6,7,8,15,17]. Kargl et al [7] present a defense mechanism that DDoS attack traffic can be reduced and limited by using Class Based Queuing (CBQ). This mechanism, however, is somewhat complex and not effective because it is needed to manage all monitored IP addresses to distinguish DDoS packets from normal ones. On the other hand, Ricciuli et al [15] randomly drop the SYN flooding packets. However, their method works only with SYN flooding attacks and drops all packets (including normal packets) when SYN flooding attacks are detected.

We present a network traffic analysis mechanism to efficiently detect DDoS attacks and probabilistically drop the suspected packets to defend DDoS attacks. To drop the suspected packets, we use a method of probabilistic packet drop such as Random Early Detection (RED) [1].

Chapter 2 shows other researches to detect and defend DDoS attacks and chapter 3 presents traffic rate analysis (TRA) to efficiently analyze network traffic under DDoS attacks. Then, the differences between Web service traffic and DDoS attack traffic using proposed TRA are explained in chapter 4. In chapter 5, the experimental results of dropping suspected packets are shown. We summarize our research and mention future work in chapter 6.

2 Related Work

An efficient management of network traffic reduce the damage caused by DDoS attacks. Accordingly, many current researches are focusing on managing network traffic [2,7,15]. Kargl et al [7] divide network bandwidth into several queues which have different network bandwidth using CBQ techniques, then classify network packets and make them flow through the classified queue in each. For instance, if normal network traffic flows through a high bandwidth queue and DDoS attack traffic flows through a queue of low bandwidth, flooding packets of DDoS attacks can be reduced. However, this defending scheme needs IP address management because packet classifying is done by seeing the IP address. Therefore, it can be said that this defending scheme is inefficient. On the other hand, Ricciuli et al [15] randomly drop a SYN flooding packet to insert a new SYN packet. This is useful to defend SYN flooding attacks, but can be applied to defend only SYN flooding attacks. In this paper, we use a packet dropping scheme like Ricciuli et al.

Detecting the DDoS attacks is an essential step to defend DDoS attacks and many researchers thus have been working on detecting the attacks [4,8,17]. Almost DDoS attackers use IP spoofing to hide their real IP addresses and locations. Since spoofed IP addresses are generated randomly, this characteristic of randomness may reveal the occurrence of DDoS attacks. Gil and Poletto [4] examine the disproportion between to-rate of the network traffic flows to a specific subnet and from-rate of the network traffic flowing from a specific subnet. This method also uses the characteristic of randomness of source IP addresses. When DDoS attacks occur, there comes a big mismatch between to-rate toward the victim and from-rate flowing to the outside from the victim. Kulkarni et al [8] presents DDoS detection method based on this characteristic of IP spoofing. This method uses Komogorov complexity metrics [11] to find randomness of source IP addresses in network packet headers. Wang et al [17] propose SYN-FIN(RST) pairs to detect SYN flooding attacks. This can be done by monitoring the ratio of SYN and FIN, but is applicable only to SYN flooding attacks.

3 Traffic Rate Analysis

Traffic rate analysis is one of network traffic analyzing methods [10]. It examines the occurrence rate of a specific type of packets within the stream of monitored network

traffic, and is composed of TCP flag rate and Protocol rate. TCP flag rate is defined in the following equation.

$$R_{td}[F i | o] = \frac{\sum \text{flag}(F) \text{ in a TCP header}}{\sum \text{TCP packets}} \quad (1)$$

TCP flag rate means the ratio of the number of a specific TCP flag to the total number of TCP packets¹. In the equation (1), a TCP flag 'F' can be one of *SYN*, *FIN*, *RST*, *ACK*, *PSH*, *URG*, and *NULL*, and 'td' is the time interval to calculate the value. The direction of network traffic is expressed as 'i' (inbound) and 'o' (outbound). For example, $R_i[SYN]$ means the occurrence rate of *SYN* flags within TCP packets when measuring inbound network traffic (toward the monitored network) during 1 second.

$$R_{td}[[TCP|UDP|ICMP]i | o] = \frac{\sum [TCP|UDP|ICMP] \text{ packets}}{\sum IP \text{ packets}} \quad (2)$$

Protocol rate is defined in equation (2). It means the ratio of a specific Layer 4 protocol (e.g. TCP, UDP, and ICMP) packets to total Layer 3 (IP) protocol packets. For instance, $R_i[TCP]$ means the occurrence rate of TCP packets within IP packets when measuring outbound network traffic (from the monitored network) during 1 second.

4 Network Traffic Analysis

In this chapter, we analyze normal Web traffic and DDoS attack traffic using proposed traffic rate analysis (TRA). A network traffic analyzer can be made using *libpcap* [9] to capture the network traffic. This analyzer is located on the adjacent site of target Web server and captures network traffic both inbound and outbound packets through Ethernet hub, then calculates TCP flag rate and protocol rate by a second.

4.1 Normal Web Service Traffic

This chapter shows the characteristics of normal Web service traffic without any DDoS attack. Actually, the characteristics of Web service traffic depends on the number of users, access pattern of the users, and their web browser. To make various network traffic of Web services, we use two Web traffic generating tools (*SPEC-web99* and *MS Web Application Stress*) [12,16]. These tools send HTTP requests to the Web server and receive HTTP replies from the Web server as the real Web browsers do.

Figure 1 shows the result of *MS Web Application Stress*. We also change network settings to make various network environments. The number of Thread (T) is changed as 1,2,3,4,5 and Sockets per Thread (S/T) as 5,10,15,20. As a result, experimental result shows a constant pattern without regard to *T* and *S/T*.

¹ The sum of calculated TCP flag rates may exceed 1.0 because a TCP packet can have one or more flags set.

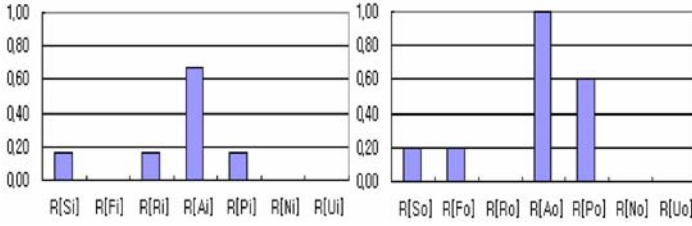


Fig. 1. Web service traffic using *MS Web Application Stress*. This also shows a specific pattern of normal Web service traffic.

As we can see in the Figure 1, *RST* packets are detected instead of *FIN* packets. This is because *MS Web Application Stress* uses *RST* packets instead of *FIN* packets to terminate TCP connections. Really, some web browsers act like this. The other differences from *SPECweb99* are the fact that $R[Si]$, $R[So]$ and $R[Fo]$ is higher than that of *SPECweb99* and $R[Ai]$ is lower than that of *SPECweb99*. The other factors are almost identical comparing with the result of *SPECweb99*.

$$\begin{aligned}
 &R[Si], R[Fi], R[Ri], R[So], R[Fo], R[Pi] \leq 0.2 \\
 &R[Ai] \leq 1.0 \\
 &R[Ao] \equiv 1.0 \\
 &R[Po] \leq 0.7 \\
 &Etc \equiv 0.0
 \end{aligned} \tag{3}$$

We examined network characteristics of normal Web service traffic by changing some network parameters using *SPECweb99* and *MS Web Application Stress*, then found that normal Web service traffic has a pattern as shown in equation 3.

4.2 DDoS Attack Traffic

We examined the characteristics of normal Web service traffic in section 4.1. In this section, we examine the change of network traffic when a Web server is attacked by various DDoS attacks.

Figure 2 shows the change of network traffic when SYN flooding attack occurs. We make Web service traffic from the time of 10 seconds to 82 seconds, and SYN flooding attack from 27 seconds to 67 seconds. The rates of *SYN* and *URG* increase almost 1.0 and the rate of *ACK* decreases almost 0.0, but other big changes do not occur.

We examined the changes of network traffic characteristics under typical DDoS attacks (SYN, UDP, ICMP flooding attacks) and could find significant differences between normal Web service traffic and DDoS attack traffic. We believe that we can early detect and defend DDoS attacks by utilizing these differences and changes of network traffic.

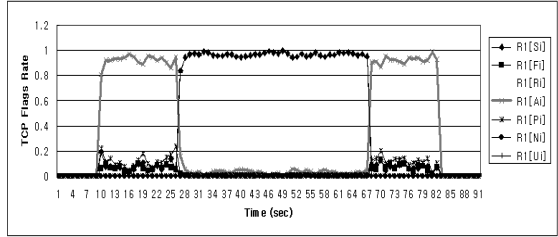


Fig. 2. SYN flooding attacks against Web servers. Under SYN flooding attacks, the rates of SYN and ACK of inbound traffic change significantly.

5 Detecting and Defending DDoS Attacks

In chapter 4, we found the fact that normal Web service traffic has a specific pattern as described in equation 3, and these patterns of network traffic would be greatly changed under DDoS attacks. This research regards the network characteristics shown in equation 3 as the standard characteristics of normal Web service traffic, and presents *Probabilistic Packet Dropping* model to defend DDoS attacks. In our DDoS defending scheme, if a specific type of network packets exceeds the standard rate, it will be probabilistically dropped. We believe this process helps us reduce the flooding packets of DDoS attacks.

Our proposed method is similar to Random Early Detection (*RED*), which is one of active queue management and used for the purpose of congestion avoidance on network router equipments [1,2]. *RED* doesn't drop the packets when average queue size is smaller than *Minimum Threshold*, drops the packets with the probability of from 0.0 to *Maximum Probability* when average queue size is greater than *Minimum Threshold* and smaller than *Maximum Threshold*, and drops all the packets if the average queue size is greater than *Maximum Threshold* [1].

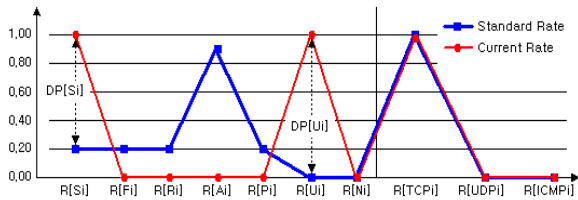


Fig. 3. Probabilistic packet dropping model. We used this model to drop suspected packets of DDoS attacks.

Figure 3 describes probabilistic packet dropping model proposed in this paper. Let the currently analyzed network traffic rate by TRA be as Current Rate, and equation 3 as Standard Rate. For example, if $R[S_i]$ and $R[U_i]$ are exceeded the standard rates in case of SYN flooding attacks, Drop Probabilities (DP) can be calculated like equation 4. Then, SYN and URG packets can be dropped with the calculated probabilities of

DP. For instance, the drop probability of *SYN* packets is 0.8 ($1.0 - 0.2$) and 1.0 ($1.0 - 0.0$) for *URG* packet in Figure 3.

This means that since the occurrence rate of *SYN* packets is 0.2 and that of *URG* packets is 0.0 in normal Web service traffic, 80% of *SYN* packets must be DDoS attack traffic and 100% of *URG* packets must be DDoS attack traffic.

$$\text{Drop Probability}(DP) = \text{Current Rate} - \text{Standard Rate} \quad (4)$$

We believe our DDoS defending scheme will help us to protect Web servers from DDoS attacks and to prove the availability of our scheme through experimental results in the next chapter.

5.1 Experimental Environment

Figure 4 shows the network settings to test our DDoS defending mechanism in a simulated environment.

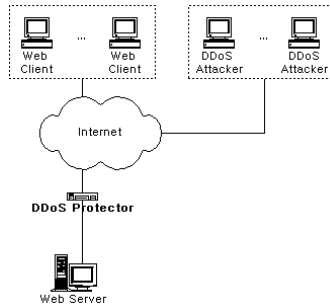


Fig. 4. Network setting. We implemented this environment using libpcap for DDoS protector, MS Application Stress for Web clients, TFN2K for DDoS attackers, and Apache for Web server.

Web clients send HTTP requests to and receive HTTP documents from the Web server using *MS Web Application Stress* [12]. While the normal Web traffic flows between Web clients and Web server, DDoS attackers generating flooding traffic against the Web server using *TFN2K* [14]. We used Linux based *Apache* for the Web server. The DDoS protector captures the network traffic both inbound and outbound one, analyze them using *TRA*, determines DPs of each packets, and finally forwards or drops the network packets. It works on the Linux 2.4.18 and uses *libpcap* to capture the network traffic and raw socket to forward packets [9].

To prove the availability of our defense mechanism, we build two different network settings: No Protection and Protection. These network settings are described in Table 1.

Table 1. Network traffic settings. We used these settings to compare protection and no protection.

	No Protection				Protection			
DDoS attack	×	SYN	UDP	ICMP	×	SYN	UDP	ICMP
Defense	×	×	×	×	○	○	○	○

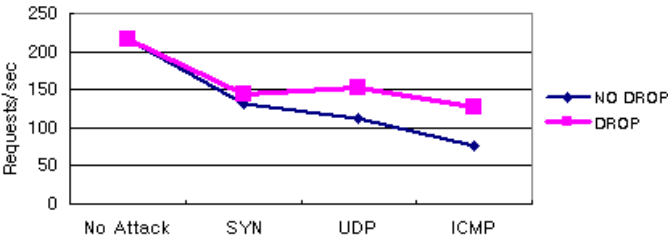


Fig. 5. Performance of our defense mechanism. Our packet dropping mechanism helps reduce the damage of DDoS attacks.

5.2 Experimental Results

Figure 5 shows the experimental results of our DDoS defense mechanism. The normal Web service traffic flows during 60 seconds, and various DDoS attacks are done between 20 seconds and 40 seconds.

As we can see in Figure 5, our defense mechanism shows high performance in defending DDoS attacks. Moreover, our defense mechanism shows higher performance in defending UDP and ICMP flooding attacks comparing with SYN flooding attacks. In SYN flooding attacks, there is only slight difference between our mechanism and no defense. It’s because some of normal *SYN* packets are dropped while dropping *SYN* packets. This is the very disadvantage of probabilistic packet dropping mechanism. Nevertheless, it can be said that our packet dropping mechanism helps minimize the performance degradation of the victim.

6 Conclusions

We presented Traffic Rate Analysis (TRA) as a network traffic analyzing method to early detect DDoS attacks and a defense scheme to protect Web servers from DDoS attacks. Our defense scheme is to probabilistically drop the suspected packets after detecting DDoS attacks via *TRA*. Experimental results showed that probabilistic packet dropping can help protect Web servers from DDoS attacks. However, the performance, especially in SYN flooding attacks, is not so much high as expected. We will focus on the methods to overcome this defect.

References

1. Braden, B., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet, *RFC 2309*, (1998).
2. Floyd, S., and Jacobson, V.: Random Early Detection (RED) gateway for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, (1993) 397-413.
3. Garber, L.: Denial-of-Service Attacks Rip the Internet, *IEEE Computer*, vol. 33(4), (2000) 12-17.
4. Gil, T.M., and Poletto, M.: MULTOPS: a data-structure for bandwidth attack detection, In *Proceedings of the 10th USENIX Security Symposium*, (2001) 23-38.
5. Houle, J.K., and Weaver, M.G.: Trends in Denial of Service Attack Technology, CERT Coordination Center, (2001).
6. Householder, A., Manion, A., Pesante, L., and Weaver, M.G.: Managing the Threat of Denial-of-Service Attacks, CERT Coordination Center, (2001).
7. Kargl, F., Maier, J., and Weber, M.: Protecting Web Servers from Distributed Denial of Service Attacks, In *Proceedings of the 10th International Conference on World Wide Web*, (2001) 514-524.
8. Kulkarni, A.B., Bush, S.F., and Evans, S.C.: Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics. Technical report 2001CRD176, GE Research and Development Center, (2001).
9. Lawrence Berkeley National Labs Network Research Group. Libpcap
10. Lee, C., Noh, S., Choi, K., and Jung, G.: Characterizing DDoS Attacks with Traffic Rate Analysis, In *Proceedings of the IADIS e-Society*, vol. 1, (2003) 81-88.
11. Li, M., and Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, Springer-Verlag, Section 7.6, (1997) 506-509.
12. Microsoft: Web Application Stress Tool v1.1, (2000). available-online: <http://www.microsoft.com/technet/itsolutions/intranet/downloads/webstres.asp>.
13. Moore, D., Voelker, G.M., and Savage, S.: Inferring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Symposium*, (2001) 9-22.
14. Packet Storm. Tribe Flood Network 2000 (TFN2K) DDoS tool, available on-line: http://packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt.
15. Ricciuli, L., Lincoln, P., and Kakkar, P.: TCP SYN Flooding Defense, *Communication Networks and Distributed Systems Modeling and Simulation*, (2000).
16. Standard Performance Evaluation Corporation. SPECweb99 Benchmark, available on-line: <http://www.spec.org/osg/web99>.
17. Wang, H., Zhang, D., and Shin, K.G.: Detecting SYN Flooding Attacks, In *Proceedings of IEEE INFOCOM – The Conference on Computer Communications*, vol. 21, no. 1, (2002) 1530-1539.

Building a Secure Infrastructure for P2P Applications in Mobile Ad Hoc Networks*

Guangming Hu, Zunguo Huang, Hua-Ping Hu, and Zhenghu Gong

School of Computer Science, National University of Defense Technology
Changsha Hunan, China
gm_hu@msn.com, zuoguo@263.net

Abstract. P2P and MANET share many similarities. Deploying P2P application in MANET faces many difficulties, especially when security consideration is incorporated. In this paper we propose a cluster-based architecture combined with a group key management scheme for mobile ad hoc network. A network is divided into clusters and each cluster has a leader called clusterhead. In order to address the problem of service registration and discovery better, we also propose a simple multicast-tree algorithm which organizes all clusterheads into trees based on each source clusterhead. The feasibility of this concept was verified by simulation results.

1 Introduction

The emergence and advances of mobile ad hoc networks (MANET) have provided a new scene for peer-to-peer systems. P2P and MANET share many similarities[1]. First of all, they both are self-organizing and decentralized, thus there is no single point of failure in P2P and MANET networks, which makes these networks as whole comparable robust and reliable. However, deploying P2P systems in MANET is not as simple as they are in wired world, where some successful systems such as Napster[2] and Gnutella[3] are being in work. It must account for the challenges produced by the nature of underlying MANET, particularly the limited resources (like computing ability, bandwidth and energy, etc.) and dynamic topology due to mobility.

The known proposed solutions to this problem are mainly focused on the information dissemination. [4] presents a P2P file sharing approach in MANET which is named Optimized Routing Independent Overlay Network (ORION). [5] introduces Konark, a service discovery and delivery protocol for P2P in mobile ad hoc networks, which uses a completely distributed, peer-to-peer mechanism that provides each peer the ability to advertise and discover resources in the ad-hoc network. From data management's perspective, [6] examines the interaction of these two self-organizing networks and presents an information dissemination paradigm.

* This work is supported in part by National 973 Program(Grant No. 2003CB314802) and National 863 Program(Grant No. 2003AA142080).

However, all the methods mentioned above do not fully consider the peculiarities of the underlying networks. The P2P overlay maintenance overhead can be substantial, thus these systems may not be applicable to mobile ad hoc networks. Furthermore, none of them take the security into consideration. In fact, due to broadcasting wireless channels and no fixed infrastructure, ad hoc networks are more vulnerable to various kinds of attacks. In this paper, we propose an approach to build secure infrastructure for P2P applications in MANET. We use secure clustering to form a virtual backbone for services discovery. Cluster-based ad hoc networks can substantially reduce the overhead for maintaining the infrastructure.

The rest of the paper is organized as follows. In section 2 we described our secure clustering concept in detail. Section 3 a simple multicast tree algorithm user to facilitate the service registration and discovery is introduced. Performance simulation and evaluation are done in section 4. Finally, in section 5, we summarize and conclude this paper.

2 Secure Clustering

The target of clustering is to construct hierarchical ad hoc networks. Inside a cluster, one node named clusterhead is in charge of coordinating the cluster activities; the nodes that can hear two or more clusterheads are called gateways; others are ordinary nodes. There are many clustering approaches [7, 8, 9]. A simple one, called the lowest-ID clustering algorithm [7]. Assuming each node has a global unique identifier (ID), the node with the lowest ID (in the neighborhood) is elected as the clusterhead. Its direct neighbors become the members of the cluster. All nodes in the cluster can hear the clusterhead, and all intra-cluster communications occur in at most two hops. While all inter-cluster communications occurs through the gateways nodes.

We take lowest-ID approach to implement our secure clustering scheme for its simplicity. Other approaches which distinguished themselves from lowest-ID' in the standard of selecting a clusterhead are also acceptable. Neither lowest-ID nor others takes security into consideration for the maintenance of clusters has already introduced additional overheads and secure-enhancing will put a heavy burden on the networks and exhaust limited network resources. To avoid this, we modified the process of maintaining to achieve localized property according to the idea examined in [10]:once a cluster is constructed, a non-CH will never challenge the current CH. If a CH moves into an existing cluster, one of the CHs will give up its role of CH based on some predefined priority.

We adopted the distributed certificate services proposed in [11] to implement the identification of messages, the authentication of authorized nodes, and the integrity of messages. In order to reduce the overhead for encrypting/decrypting, we produce the same communication key for each authorized node using the distributed group key management framework proposed in [12]. The group key management framework in [12] is also based on a shared certification key and threshold cryptography idea for securing ad hoc network that was first presented by Zhou and Haas [13]. It uses an offline controller to set keys for each nodes before deploying the network. The

communication key is divided into n parts amongst all network nodes and any k nodes can reconstruct the key while $k-1$ or fewer nodes cannot.

2.1 Network Model and Definitions

We model an ad hoc network by an undirected graph $G = (V, E)$ in which V is the set of nodes and there is an edge $\{u, v\} \in E$ if and only if u and v can mutually receive each others' transmission. The rest of the notation and the definitions is as follows.

N : Set of all the nodes in the network.

$N(i)$: Set of the neighbors of node i .

$M(i)$: Set of the members of a cluster, initially nil. It is updated if and only if i is a CH.

U : Set of undecided nodes.

ch_i : Clusterhead i .

ID_i : Network identifier for node i .

T_h : HELLO beacon periods.

T_w : Waiting time, usually $T_w > 2 * T_h$.

2.2 Message-Driven Algorithm

Except for the initiation, the secure LID algorithm is driven by messages, namely, each nodes runs special procedure according to arriving messages. The following messages are needed:

- $HELLO_i$: The node i in Undecided state broadcasts HELLO message to its neighbors periodically. A typical HELLO message is as follows:

$HELLO_i = (ID_i \mid hello \mid timestamp \mid sk_i(hash))$

ID_i is node i 's ID, message type is hello. We do not require there is a global synchronous clock, so we use logic clock to timestamp the message for implementing asynchronous communication. $sk_i(hash)$ means to sign the abstract of the message using node i 's private key sk_i .

- $JOIN(v, u)$: When node u wants to join cluster v , it broadcasts the message periodically. A typical JOIN message is as follows:

$JOIN_u = (ID_u \mid member \mid timestamp \mid sk_u(hash))$

Except the message type is member, other items are same as that of HELLO.

- $CH(v)$: Node v periodically broadcasts this message when it becomes a clusterhead. In our scheme, the format of $CH(v)$ is:

$CH(v) = (ID_v \mid clusterhead \mid timestamp \mid sk_v(hash))$

- **RESIGN(u)**: When two clusterheads meet each other, one of which gives up its identity. The message is as follows:

$$RESIGN(u) = (ID_u \mid resign \mid timestamp \mid sk_u(hash))$$

All messages are encrypted with the global group communication key to keep their confidentiality. The algorithm works as follows.

Initially, each node is undecided. The ch_i of node i is nil and the $N(i)$ is Φ . Each node periodically broadcasts its HELLO messages to all its neighbors. When a message arrives, a node v will verify its source and integrity according its signature, then modify its neighbor set $N(v)$. After T_w , if node i is Undecided ($ID_i \in U$) and its ID is the lowest compared to all its neighbors, then node i becomes clusterhead and broadcasts $CH(i)$ messages to its neighbors periodically.

When an undecided node j receives the $CH(i)$ message, it first verifies if the message comes from an authorized node. If node j 's ID is larger than that of node i ($ID_j > ID_i$), then node j joins the cluster i .

2.3 Security Analysis

In our secure Lowest-ID algorithm, each message for clustering is first signed with the node's private key, then is encrypted with the global group communication key. Since unauthorized malicious nodes cannot gain the group communication key and certificates, they are not able to destroy the clustering process by forging false messages. Similarly, neither malicious node can destroy the clustering process by modifying messages.

The situation that a node is compromised is some more complex. The only useful information a compromised node wants to forge is its ID. Since a node's ID is global unique in assumption, and during the period of key distribution, the ID is built into the node's group member certificate, it is nearly impossible to forge a false ID. A compromised node can send false JOIN message, but this only affects itself and will not do harm to the clustering process.

3 Services Registry and Discovery

In order to enable a peer to locate others' services and in order to make a peer's services available to other peers, we have each clusterhead to act as a directory agent (DA), which processes the request from clients and the registry from servers in the network. Directory-less architecture does not have any directory agent and seems more suitable for the nature of mobile ad hoc networks, but a service query request needs to be flooded in the whole network, which may cause too much traffic, so it is abandoned in our scheme. All clusterheads form a virtual network (we call it CH-network), the links in which are consisted of physical links and gateway(s). Any node in a cluster wants to register its service has to register with the DA on its clusterhead.

Service registry is not strictly confined within a cluster. A node can register its service with more DAs on other clusterheads. When a node changes its clusterhead due to mobility or other causes, it must register its service with new clusterhead. The registry information with the original DA will be eliminated when time is out. The query for a service is similar to the registry process of a service. A node sends request to the DA on its clusterhead. If there is no entry for that service, the node will query it from other DAs using multicasting or broadcasting.

To make the services registry and discovery more efficient, we propose a simple clusterhead-based multicast tree algorithm. Each clusterhead identifies a tree. The forming of a tree is initiated by service discovery request or registration messages sent to the clusterhead by any node in that cluster.

Each clusterhead keeps a service-forwarding-list for each multicast tree. Initially, each service-forwarding-list only include all its direct neighbor clusterheads. When a node (whether clusterhead or non-clusterhead) wants to register its service, it has to send a SERVICE_REGISTRATION message to the DA located on its clusterhead. A SERVICE_REGISTRATION message may contain the service name, type, URL, TTL, etc, and will be organized by the DA in any forms. Then the clusterhead will send the registration message to all of its neighbor clusterheads according its service-forwarding-list. If a neighbor clusterhead receives the registration message for the first time, it will forward the message to its own direct neighbor clusterheads. If CH_A receives the replicas of the message from its neighbor clusterheads, it will delete the relative clusterheads from its service-forwarding-list. Finally, the multicast tree identified by root clusterhead is formed on top of the CH-network.

The formation of a multicast tree can also be triggered by service discovery request message. Its procedure is similar to that of the service registration.

To avoid the registration messages crowd the network, a TTL field should be set in the message. When a registration message travels a clusterhead, its TTL should subtract 1. If the TTL equals zero, the message will not be forwarded by any clusterhead.

4 Simulation and Evaluation

In this section, we first describe the performance metrics that are used to evaluate our secure clustering algorithm as well as service registration and discovery algorithm. Then the simulation scenarios and simulation results are presented.

4.1 Performance Metrics

Three performance criteria are considered in our simulations. The first performance metric is the stability of the CH-network that can be reflected by the changing frequency of clusterheads. The second performance metric is the control message overheads that measure the load of the algorithms on network resources in terms of the number of packets, especially the overhead caused by our security enhancement. The last performance metric is the average delay between the time any successful request is sent from a client and the time corresponding reply is received by the same client.

4.2 Simulation Scenarios

We simulated our secure clustering algorithm and service discovery mechanisms using random waypoint mobility model in ns-2 [14]. The random waypoint mobility model is the most widely used one for the performance evaluation of various ad hoc networking aspects.

We implemented our secure Lowest-ID(SLID) algorithm using an extended agent, and other two clustering algorithms (Highest-Degree and WCA) are also implemented for comparison. We generate different scenarios by randomly placing 50 nodes in a $1000m \times 1000m$ square with transmission power ranging from 30m to 180m. The pause time between two continuous movements in this model is set to 4s and the speed of a node varies from 0 to 50m/s. Each undecided node sends its HELLO messages every $2s(T_h=2)$, and if two nodes have no any messages exchanged for more than $4.2s(T_w=4.2)$, then the link between them is believed unavailable. The total simulation time is 600s.

4.3 Simulation Results

In the first set of experiments, we want to find the impact on clusterheads' stability when the nodes' moving speed or transmission range varies.

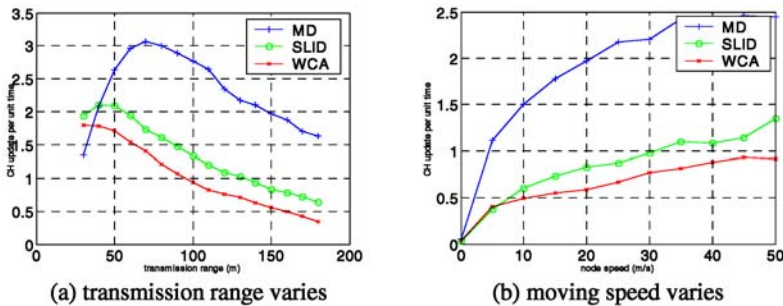


Fig. 1. Clusterheads updating frequency when transmission range and nodes' speed varies.

Fig 1(a) shows the updating frequency of clusterheads when the transmission range varies. We can see that the SLID algorithm is not as good as WCA but better than Highest-Degree algorithms, and when the transmission range is more than 90m, the updating frequency decreases along a increasing transmission range. From fig 1(b) we can see the stability of SLID algorithm is close to WCA when nodes' speed increases, but far more better than Highest-Degree algorithm.

In the second set of simulations, we want to know if the control message overheads caused by security-enhancing will be a heavy burdens on the network. Fig 2 shows that overheads in SLID algorithm are not a significant impact to the network, and the control packets decrease with the increase of transmission range.

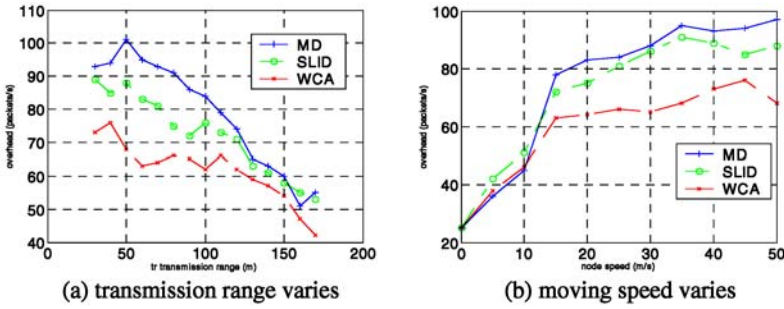


Fig. 2. The overheads in unit time when transmission range and nodes' speed vary.

The last set of simulations is for verifying the performance multicast-tree algorithm on different underlying clustered architecture. We take the following assumptions: 20 clients request services, but there are only 3 servers responding. Fig 3 shows the delay in SLID algorithm is middle among these three underlying clustering algorithms. From these three sets of experiments we think our secure enhancement scheme for building an ad hoc infrastructure for P2P applications is acceptable.

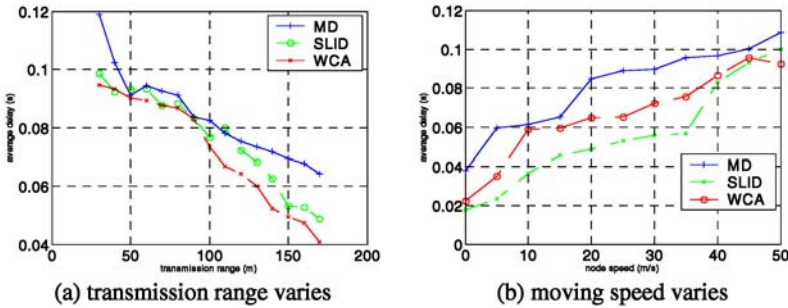


Fig. 3. The average delay when transmission range and nodes' speed vary.

5 Conclusion

In this paper, we examined how to build a secure infrastructure for P2P applications in mobile ad hoc networks. P2P and MANET are both self-organizing and decentralized, but MANET has some peculiarities. In order to adapt to the nature of MANET, we divided the ad hoc network into clusters. We use a secret sharing and threshold cryptography scheme to enhance the security of the clustered architecture, where each node holds a piece of the global communication key, and any k nodes can reconstruct it. All clusterheads form a virtual network where the directory agents resided in. In order to facilitate the service registration and discovery, we put forward a simple multicast algorithm. Due to heavy overheads, we do not consider the encryption of the packets for service registration and discovery.

In order to evaluate our approach, we examined the stability of the clusterheads network and the amount of control overheads based on the widely used random waypoint mobility model. We also showed how a node's speed and transmission range variations affect the performance.

In this paper, we have not investigated the more complex security level and access control – a valid node now has full access authorization, nor have we investigated how to use traditional service discovery techniques like 'PUSH' or 'PULL' to improve the system performance. Our future research should address these problems.

References

1. Rüdiger Schollmeier, Ingo Gruber, and Michael Finkenzeller. Routing in mobile ad hoc and peer-to-peer networks. A comparison. International Workshop on Peer-to-Peer Computing, Italy, May 2002
2. Napster Homepage. Napster. <http://www.napster.com/>.
3. Gnutella Homepage. Gnutella . <http://www.gnutella.com/>.
4. A. Klemm, C. Lindemann, and O. Waldhorst, A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks, Proc. IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL, October 2003.
5. Desai B., Verma V., Helal S., "Infrastructure for Peer-to-Peer Applications in Ad-Hoc Networks", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003.
6. Anwitaman Datta, MobiGrid : P2P Overlay and MANET Rendezvous -a Data Management Perspective, CAiSE 2003 Doctoral Symposium, in conjunction with the 15th Conference On Advanced Information Systems Engineering, Klagenfurt/Velden, Austria, 16 - 20 June, 2003.
7. M. Gerla and J. Tzu-Chieh Tsai, Multiclust mobile, multimedia radio network, Wireless Networks, Vol. 1, pp. 255-265, Oct. 1995.
8. Ching-Chuan Chiang and Mario Gerla. "Routing and Multicast in Multihop, Mobile Wireless Networks." In IEEE International Conference on Universal Personal Communications (ICUPC'97), pp. 546–551, 1997.
9. M. Chatterjee, S.K. Das and D. Turgut. "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", Journal of Clustering Computing IEEE Vol. 5, No. 2, April 2002 pp.193-204
10. Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla, Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel, In Proceedings of IEEE Singapore International Conference on Networks, 1996, 197-211
11. J Kong, P Zeros, H. Luo, S Lu and L Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In Ninth International Conference on Network Protocols (ICNP'01): 251~260, 2001.
12. Xiao-Hui Kuang. Research of group key management in mobile ad hoc networks. PhD thesis, National University of Defense Technology, Changsha, Hunan, P.R. China, 2003.
13. L. Zhou and Z. J. Haas, "Securing ad hoc networks," IEEE Network, vol. 13, no. 6, pp. 24-30, 1999.
14. The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>, 2002.

Middleware Framework for Secure Grid Application in Mobile Web Services Environment

Namje Park, Kiyoun Moon, Jongsu Jang, and Sungwon Sohn

Information Security Research Division, ETRI
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{namejepark, kymoon, jsjang, swsohn}@etri.re.kr

Abstract. Mobile Grid Service is the extension of Grid Service. It is defined as: it is an intelligent code service wandering in grid nodes to accomplish certain task and provide certain service. Mobile Grid Service provides a series of standard interfaces and conforms specific conventions to solve such problems as: mobile service discovery, dynamic service creation, lifetime management, notification, mobile service interacting and mobile service migration, etc. The goal of this paper is to investigate how well the most limited wireless devices can make use of Grid Security Services. This paper describes a novel security approach on Mobile Grid Services to validate certificate based on current Mobile Web Services platform environment using XML Security mechanism.

1 Introduction

Grid Computing emerges as a technology for coordinated large-scale resource sharing and problem solving among many autonomous groups. In Grid's resource model, the resource sharing relationships among virtual organizations are dynamic. However, Grid requires a stable quality of service provided by virtual organizations and the changing of sharing relationship can never happen frequently. This model works for a conventional distributed environment but is challenged in the highly variational wireless mobile environment[3]. Besides Mobile Internet the traditional Internet computing is experiencing a conceptual shift from Client-Server model to Grid and Peer-to-Peer computing models. As these trends, Mobile Internet and the Grid, are likely to find each other the resource constraints that Wireless devices pose today affect the level of interoperability between them[2].

Grid is the umbrella that covers many of today's distributed computing technologies. Grid technology attempts to support flexible, secure, coordinated information sharing among dynamic collections of individuals, institutions, and resources. This includes data sharing but also access to computers, software and devices required by computation and data-rich collaborative problem solving. So far the use of Grid services has required a modern workstation, specialized software installed locally and expert intervention. In the future these requirements should diminish considerably. One reason is the emergence of Grid Portals as gateways to the Grid. Another reason is the 'Web Service' boom in the industry. The use of XML as a network protocol

and an integration tool will ensure that future Grid peer could be a simple wireless device[2,3].

Furthermore, open Mobile Grid service infrastructure will extend use of the Grid technology or services up to business area using Web Services technology. Therefore differential resource access is a necessary operation for users to share their resources securely and willingly. Therefore, this paper describes a novel security approach on open Mobile Grid service to validate certificate based on current Mobile Grid environment using XKMS (XML Key Management Specification) and SAML (Security Assertion Markup Language), XACML (eXtensible Access Control Markup Language) in XML (eXtensible Markup Language) security mechanism.

This paper is organized as follows. First we investigate related work on Mobile Grid and mobile web services. Then we propose a design of security system platform for open mobile Grid service and explain experimented XML-based Key Management System model for certificate validation service. Finally, we explain function of system and then we conclude this paper.

2 Mobile Grid Computing Based on Mobile Web Services

The Open Mobile Alliance (OMA) this week released Mobile Web Services, which defines best practices by which mobile applications can be exposed, discovered, and consumed using Web services[9]. The technology supports a business and technology model for Web services using open standards. The purposed of Mobile Web Services specification is twofold. First, to provide specifications and guidelines for Web services technologies to integrate and interoperate within the mobile architecture; and secondly, to ensure interoperability across servers and terminals supporting Web services protocols through the use of standardized protocols. In this section, we present possible architecture of the mobile grid system and several technical issues that are to be dealt with in further researches. We depict an expected view of mobile grid system in figure 1.

As illustrated in the figure, the grid system is divided into three parts: static grid sites, a group of mobile devices, and a gateway interconnecting static and mobile resources. The mobile networks allow wireless devices to become servers to peers. Wireless peers can provide content, network traffic routing, and many other services. The mobile network truly leverages wireless networks' dynamic nature. However, because wireless peer-to-peer technology is still embryonic, its many performance and security issues must be solved before it can be widely used.

3 Middleware Framework for Secure Mobile Grid Service

Web services can be used to provide mobile security solutions by standardizing and integrating leading security solutions using XML messaging. XML messaging is referred to as the leading choice for a wireless communication protocol and there are security protocols for mobile applications based upon it. Among them are the follows.

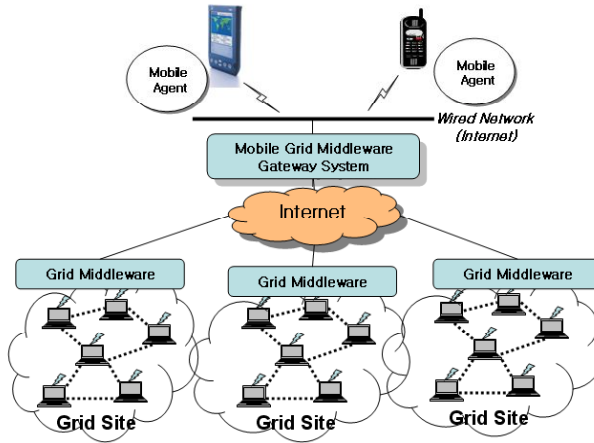


Fig. 1. Mobile Grid Services.

SAML is a protocol to transport authentication and authorization information in an XML message. It could be used to provide single sign on web services. XML signatures define how to digitally sign part or all of an XML document to guarantee data integrity. The public key distributed with XML signatures can be wrapped in XKMS formats. XML encryption allows applications to encrypt part or all of an XML document using references to pre-agreed symmetric keys. The WS-Security, endorsed by IBM and Microsoft, is a complete solution to provide security to web services. It is based on XML signatures, XML encryption, and an authentication and authorization scheme similar to SAML. When a mobile device client requests access to a back-end application, it sends authentication information to the issuing authority. The issuing authority can then send a positive or negative authentication assertion depending upon the credentials presented by the mobile device client. While the user still has a session with the mobile applications, the issuing authority can use the earlier reference to send an authentication assertion stating that the user was, in fact, authenticated by a particular method at a specific time. As mentioned earlier, location-based authentication can be done at regular time intervals, which means that the issuing authority gives out location-based assertions periodically as long as the user credentials make for a positive authentication.

CVM (Certificate Validation Module) in XKMS system perform path validation on a certificate chain according to the local policy and with local PKI (Public Key Infrastructure) facilities, such as certificate revocation (CRLs) or through an OCSP (Online Certificates Status Protocol)[4,5,6,7]. In the CVM, a number of protocols (OCSP, SCVP, and LDAP) are used for the service of certificate validation. For processing the XML client request, certificate validation service from OCSP, LDAP (Lightweight Directory Access Protocol), SCVP (Simple Certificate Validation Protocol) protocols in XKMS based on PKI are used[1]. The XKMS client generates an 'XKMS validate' request. This is essentially asking the XKMS server to go and find

out the status of the server's certificate. The XKMS server receives this request and performs a series of validation tasks e.g. X.509 certificate path validation. Certificate status is determined. XKMS server replies to client application with status of the server's certificate and application acts accordingly. Using the OCSP protocol, the CVM obtained certificate status information from other OCSP responders or other CVMs. Using the LDAP protocol, the CVM fetched CRL (Certificate Revocation List) from the repository. And CA (Certificate Authority) database connection protocol (CVMP;CVM Protocol) is used for the purpose of that the server obtains real-time certificate status information from CAs. The client uses OCSP and SCVP. With XKMS, all of these functions are performed by the XKMS server component. Thus, there is no need for LDAP, OCSP and other registration functionality in the client application itself.

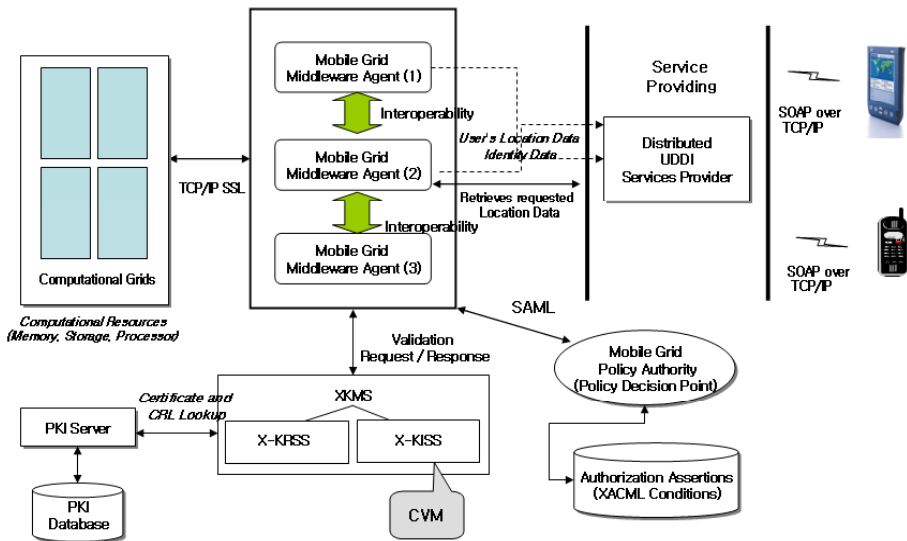


Fig. 2. Security Framework for Open Mobile Grid Middleware.

4 Protocol for Secure Mobile Grid Application

Three types of principals are involved in our protocol: Mobile Grid application (server/client), SAML processor, and XKMS server (including PKI). Proposed invocation process for secure Mobile Grid security service consists of two parts: initialization protocol and invocation protocol. The initialization protocol is prerequisite for invoking Grid web services securely. Through the initialization protocol, all principals in our protocol set up security environments for their web services, as shown in fig. 3. The flow of setting up security environments is as follows.

The client first registers its information for using web services, and then gets its id/password that will be used for verifying its identity when it calls web services via secure channel. Then, the client gets SAML assertions and installs security module to

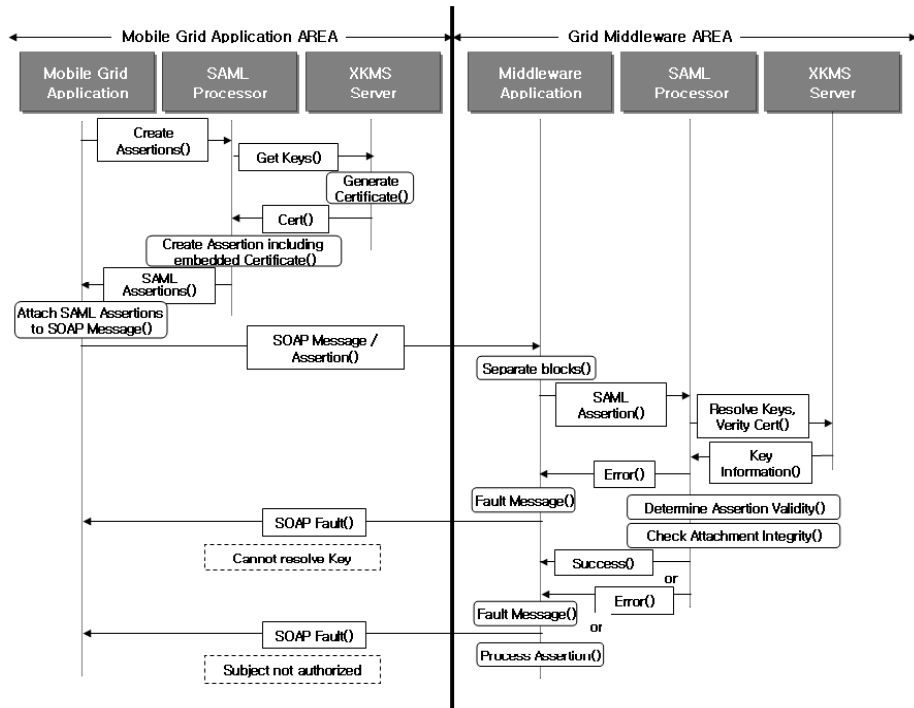


Fig. 3. Security Protocol for Secure Open Mobile Grid Service.

configure its security environments and to make a secure SOAP message. It then generates a key pair for digital signature, and registers its public key to a CA.

The client creates a SOAP message, containing authentication information, method information, and XML signature, XML encrypts it, and then sends it to a server. The message is in following form: $Enc_{session}(Envelope(Header(SecurityParameters, Sig_{client}(Body))) + Body(Method, Parameters)))$, where $Sig_x(y)$ denotes the result of applying x 's private key function (that is, the signature generation function) to y . The protocol shown in fig. 3 shows the use of end-to-end bulk encryption[12,13,16]. The security handlers in server receive the message, decrypt it, and translate it by referencing security parameters in the SOAP header. To verify the validity of the SOAP message and authenticity of the client, the server first examines the validity of the client's public key using XKMS. If the public key is valid, the server receives it from CA and verifies the signature. The server invokes web services after completion of examining the security of the SOAP message. It creates a SOAP message, which contains result, signature, and other security parameters. Then, it encrypts the message using a session key and sends it back to the client. Lastly, the client examines the validity of the SOAP message and server, and then receives the result[14,15].

In current Grid service, there is no mechanism of differential resource access. To establish such a security system we are seeking, a standardized policy mechanism is required. We employ the XACML specification to establish the resource policy

mechanism that assigns differential policy to each resource (or service). SAML also has the policy mechanism while XACML provides very flexible policy mechanism enough to apply to any resource type. For our implementing model, SAML provides a standardized method to exchange the authentication and authorization information securely by creating assertions from output of XKMS (e.g. assertion validation service in XKMS). XACML replaces the policy part of SAML as shown in fig 4.

Once the three assertions are created and sent to the protected resource, there is no more verification of the authentication and authorization at the visiting site. This, SSO (Single Sign-On), is a main contribution of SAML in distributed security systems.

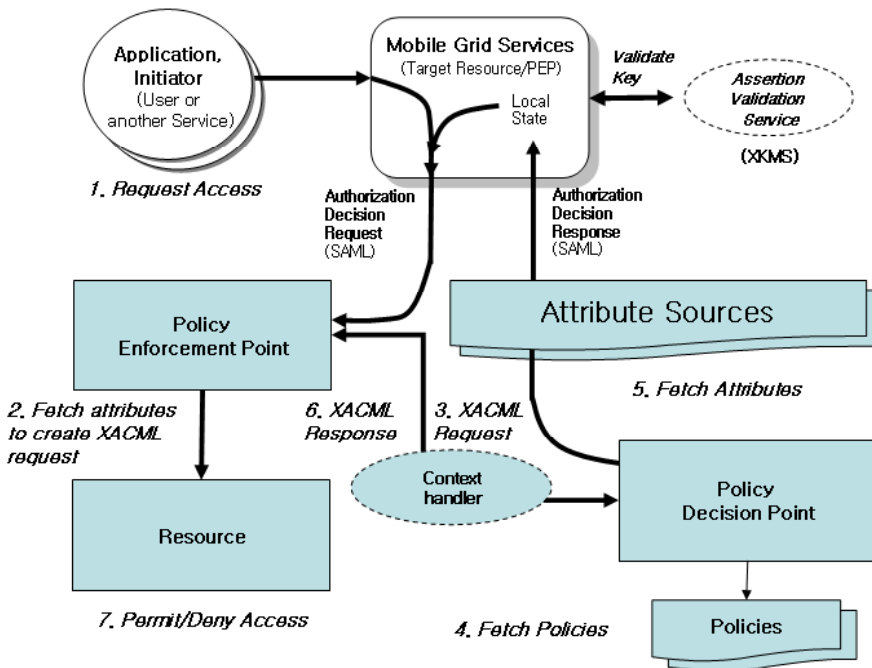


Fig. 4. Security Message Flow using SAML/XACML in Open Mobile Grid Middleware.

Fig. 4 shows the flow of SAML and XACML integration for differential resource access. Once assertions are done from secure identification of the PKI trusted service, send the access request to the policy enforcement point (PEP) server (or agent) and send to the context handler. Context handler parses the attribute query and sends it to PIP (policy information point) agent. The PIP gathers the subject, resource and environment attributes from local policy file, and the context handler gives the required target resource value, attribute and resource value to PDP (policy decision point) agent. Finally, the PDP decides access possibility and send context handler so that PEP agent allow or deny the request[10,11,13].

5 Secure Key Management of Mobile Grid Application

XKMS has been implemented based on the design described in previous section. Package library architecture of XKMS based on CAPI (Cryptographic Application Programming Interface) is illustrated in figure 5.

Components of the XKMS are XML security library, service components API, application program. Although XKMS service component is intended to support XML applications, it can also be used in order environments where the same management and deployment benefits are achievable. XKMS has been implemented in Java and it runs on JDK (Java Development Kit) ver. 1.3 or more.

The figure for representing Testbed architecture of XKMS service component is as follows fig. 5. We use Testbed system of windows PC environment to simulate the processing of various service protocols. The protocols have been tested on pentium 3 and pentium 4 PCs. It has been tested on windows 2000 server, windows XP.

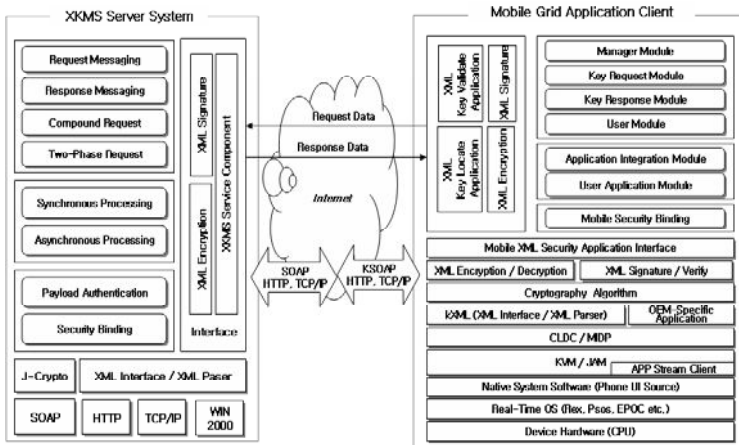


Fig. 5. Design of XKMS Component for Open Mobile Grid Services.

Java 2, Micro Edition (J2ME) is a set of technologies and specifications developed for small devices like smart cards, pagers, mobile phones, and set-top boxes. J2ME uses subset of Java 2, Standard Edition (J2SE) components, like smaller virtual machines and leaner APIs. J2ME has categorized wireless devices and their capabilities into profiles: MIDP, PDA and Personal. MIDP and PDA profiles are targeted for handhelds and Personal profile for networked consumer electronic and embedded devices[2]. As the technology progresses in quantum leaps any strict categorization is under threat to become obsolete. It is already seen that J2ME Personal profile are being used in high-end PDAs such as PocketPCs and Mobile Communicators. We will concentrate on the most limited category of wireless J2ME devices that use Mobile Information Device Profile (MIDP). Applications that these devices understand are Midlets. Typically maximum size of a midlet varies from 30-50kbs and user can download four to six applications to his mobile phone. Midlet is a JAR-archive conforming to the Midlet content specification[2].

The XKMS server is composed server service component of XKMS platform package. And the message format is based on Specification[1] of W3C (World Wide Web Consortium).

6 Conclusion

We propose a novel security approach on open Mobile Grid to validate certificate based on current Grid security environment using XML security mechanism. This service model allows a client to offload certificate handling to the server and enable to provide central administration of XKMS policies. In order to obtain timely certificate status information, the server uses several methods such as CRL, OCSP etc. Our approach will be a model for the future security system that offers security of open Mobile Grid security.

References

1. XML Key Management Specification Version 2.0 (W3C Working Draft), April 2003.
2. Miika Tuisku, Wireless Java-enabled MIDP devices as peers in Grid infrastructure, Helsinki Institute of Physics, CERN.
3. Ye Wen, Mobile Grid Major area examination, University of California, 2002.
4. E. Faldella and M. Prandini, A Novel Approach to On-Line Status Authentication of Public Key Certificates, in Proc. the 16th Annual Computer Security Applications Conference, Dec 2000.
5. Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman and S. Proctor, Building Certification Paths: Forward vs. Reverse, Proc. the Network and Distributed System Security Symposium Conference, 2001.
6. M. Naor and K. Nissim, Certificate Revocation and Certificate Update, IEEE Journal on Selected Areas in Communications, 18 (4) (2000).
7. Jonghyuk Roh et. Al., Certificate Path Construction and Validation in CVS, KICS-Korea IT Forum, 2002.
8. M. Prandini, Efficient Certificate Status Handling within PKIs: an Application to Public Administration Services, in Proc. the 15th Annual Computer Security Applications Conference, 1999.
9. Sangmin Park, et. Al., Disconnected Operation Service in Mobile Grid Computing, ISOC 2003, 2003.
10. Namje Park et. Al., Certificate Validation Scheme of Open Grid Service Usage XKMS, GCC 2003, 2003.
11. Euinam Huh, Jihye Kim, Hyeju Kim, Kiyoun Moon, Policy based on Grid security infrastructure implementation for differential resource access, ISOC 2003, 2003.
12. Yuichi Nakamura, et. Al., Toward the Integration of web services security on enterprise environments, IEEE SAINT '02, 2002.
13. Donghoon Lee, et. Al., Improved Scalar Multiplication on Elliptic Curves Defined over F_2^{mn} , ETRI Journal, V.26, n.3, June 2004.
14. Jose L. Munoz et. Al., Using OCSP to Secure Certificate-Using transactions in M-Commerce. LNCS 2846 (2003) 280-292.
15. Sungmin Lee et. Al., TY*SecureWS: An integrated Web Service Security Solution based on java, LNCS 2738 (2003) 186-195
16. Sangsu Lee, et. Al., Visual Cryptography based on an Interferometric Encryption Technique, ETRI Journal, V.24, n.5, Oct 2002.

Autonomic Computing for Defense-in-Depth Information Assurance: Architecture and a Case Study*

Xin Xu^{1,2}, Zunguo Huang¹, and Lei Xuan¹

¹ School of Computer, National University of Defense Technology
410073, Changsha, P.R. China
xuxin_mail@263.net

² Institute of Automation, National University of Defense Technology
410073, Changsha, P.R. China

Abstract. In recent years, defense-in-depth information assurance is one of the main focuses in information security research. However, the complexity of information assurance systems increases rapidly with more and more security functions and subsystems being included. In this paper, we propose an autonomic computing architecture for defense-in-depth information assurance systems (DDIAS) so that the increasing complexity of DDIAS can be tackled by distributed autonomous security subsystems with the abilities of self-configuration, self-optimization, self-healing and self-protection. We also present a case study of autonomic computing for distributed emergency response and incident recovery, which is usually the last line of in-depth defense. In the case study, we combine the tenure duty method (TDM) with autonomic system architecture to realize autonomic service roaming and dynamic backup. Experiments show that the proposed method greatly improves the survivability of information systems without much loss of quality of service.

1 Introduction

Computer security is a field that has gained significance over the past few years, especially with the widespread internetworking of computers. Until now, lots of theories and techniques have been proposed in the area of computer security, whose development can be mainly divided into two stages. The first stage is focused on static defense theory and technologies, which include cryptography, access control and firewalls. Although static defense methods install the first wall to intruders, they are hardly able to tackle with dynamic network intrusion behaviors and increasing security problems in operating systems and software. The second stage of computer security research develops different kinds of dynamic defense techniques including intrusion detection [1], honey-pots, emergency response and incident recovery [2], etc. These methods can provide more defense lines to intruders and make information systems become more flexible and survivable to dynamic environments. To integrate different kinds of defense techniques and systems, multi-layer defense-in-depth in-

* Supported by the National Natural Science Foundation of China Under Grant 60303012, and National High-Technology (863) Program under Grant 2003AA2080.

formation assurance has become an important topic in recent years [3-4], which emphasizes the significance of defense-in-depth and active protection of information systems. However, in the research and implementation of defense-in-depth information assurance systems (DDIAS), the complexity of information security systems increases with more and more defense functions and subsystems being incorporated and the system scale expanded. To manage this complexity is one of the key problems for the successful applications of DDIAS.

Autonomic computing was recently proposed as a way to develop self-managing software products [5-6]. The aim of autonomic computing is to cope with the rapidly growing complexity of operating, managing, and integrating computing systems. To realize this objective, an autonomic computing system should have four fundamental features, i.e., self-configuration, self-healing, self-optimization and self-protection. Meeting the grand challenge of autonomic computing will involve researchers in a diverse array of fields, including systems management, distributed computing, networking, operations research, artificial intelligence, and control theory, as well as others.

Although there have been noticeable advances in the research and implementation of autonomic computing systems, little work has been done on the self-management of information assurance systems to cope with the complexity of DDIAS. In this paper, we propose an autonomic computing architecture for DDIAS, where multiple loops of information feedback are designed to realize the goal of autonomic computing. To illustrate the principle of self-management based on autonomic computing, we also present a case study of autonomic emergency response and incident recovery, where a distributed autonomic service roaming scenario is considered. Simulation results on a TCP service application show that the distributed autonomic service roaming can improve the survivability of information systems without much loss of quality of service.

The paper is organized as follows. Section 2 presents a brief discussion of related work. In Section 3, the autonomic computing architecture for DDIAS is proposed. In Section 4, an autonomic distributed incident recovery system is presented as a case study of autonomic computing for DDIAS. Section 5 draws conclusions and discusses future work.

2 Related Work

The notion of information assurance (IA) was early proposed in IATF [4]. According to IATF, assurance is achieved when information and information systems are protected against attacks through the application of security services such as: availability, integrity, authentication, confidentiality, and non-repudiation. The application of these services should be based on the protecting, detection, and reaction paradigm. This means that in addition to incorporating protection mechanisms, organizations need to predict attacks and include attack detection tools and procedures that allow them to react to and recover from these attacks. All these kinds of defense mechanisms form a defense-in-depth architecture. In [9], defense-in-depth information assurance is further associated with risk management and it is illustrated that the risk evaluation and management process an organization selects is the key to building a

successful and cost-effective defense-in-depth IA strategy. The discussion in [9] extends the technical defense-in-depth boundary protection construct to a uniform qualitative risk management perspective that is tightly coupled with network implementation, resources, mission criticality, security policies and network-centric mission operations. Although the research work in this paper will focus on technical perspectives of DDIA, risk evaluation will also be considered in our autonomic computing architecture for DDIAS, where the whole DDIAS's state and performance feedback is realized by autonomic risk evaluation systems. Thus, the overall DDIAS becomes an integrated autonomic computing system with different kinds of autonomic subsystems.

After P. Horn proposed the idea of autonomic computing [8], many researchers have studied different aspects of autonomic computing principle in different areas. In [6], management issues related to topology, service placement, cost and service metrics, as well as dynamic administration structure are explored. In [7], autonomic computing for personal computing was studied. Although security problem is an important perspective in autonomic computing, how to integrate various security techniques with autonomic computing remains an open problem. Furthermore, the increasing complexity of DDIAS needs to be managed by new computing architecture and principles. This paper makes contributions in this direction by proposing an autonomic computing architecture for DDIAS and presenting a case study to illustrate the basic principles of the idea. The results in this paper can be viewed as a first step to combine the research in autonomic computing and DDIA, which may greatly promote the development of information security and IT system management.

3 Autonomic Computing Architecture for DDIA

The DDIA studied in this paper is based on the integration of various defense functions, which may include access control, intrusion detection, incident recovery and reaction, and risk evaluation, etc. In the following discussion, we will mainly focus on real-time dynamic defense functions in DDIAS.

Among the real-time defense functions in DDIAS, access control, including firewall, identity authentication, etc., serves as the first line of defense. Intrusion detection systems (IDS) are the core part of dynamic DDIA, which is the second line of defense. Incident recovery and reaction can be viewed as the last resort under intrusions. When the three layers of defense techniques are integrated in DDIAS, the damage caused by intrusions can be minimized. However, when more and more defense functions and subsystems are incorporated, the complexity of DDIAS will increase rapidly and the management of DDIAS will become a challenging problem. To solve this problem, in the following, we will present an autonomic computing architecture for DDIAS, where multiple feedback loops are introduced in different layers of DDIA and risk evaluation serves as the main loop of the whole IA system.

Fig.1 shows a basic autonomic computing architecture for DDIA, where multiple loops of feedback are introduced in every local defense layer as well as from protected critical systems to every layer. The information feedback loops enable local defense subsystems to be autonomic, which means that a subsystem can perform autonomic planning, configuration, optimization, and self-protection. In addition, the

feedback from ultimate protected information systems provides information of every subsystems' common goal, i.e., to minimize the risk of critical information systems. This information feedback provides the basis for the distributed cooperation among multiple layer defense subsystems. Another information feedback for distributed autonomic cooperation is based on risk evaluation of the whole DDIAS, which is the global and long-term evaluation about the dynamic properties of DDIAS.

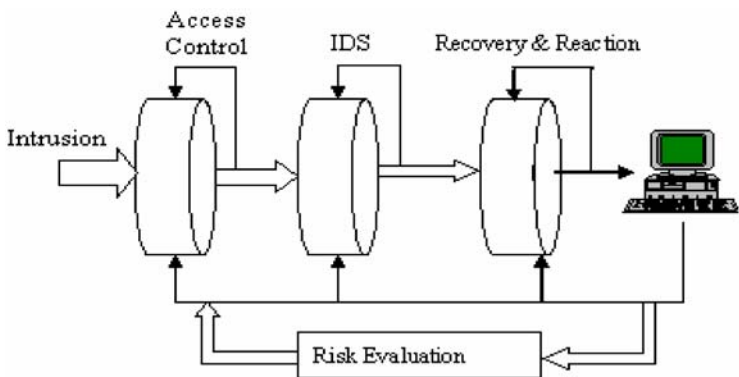


Fig. 1. Multi-loop feedback in Autonomic DDIA

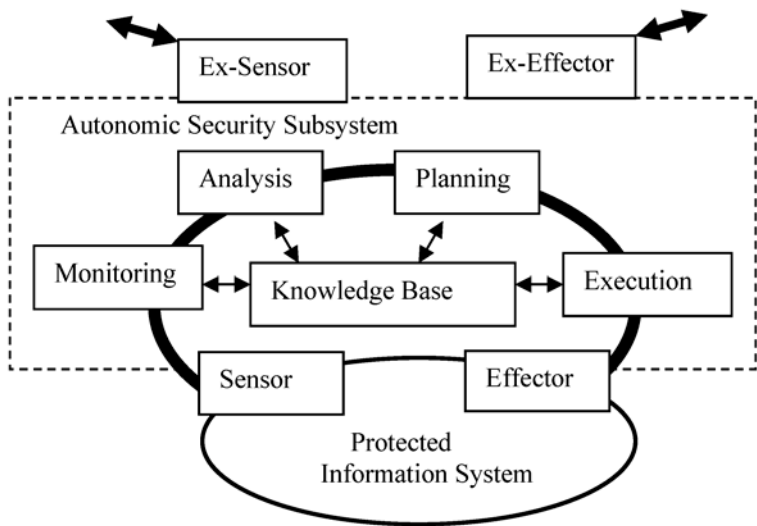


Fig. 2. Structure of an autonomic defense subsystem

The multi-loop feedback in the autonomic computing architecture for DDIA makes use of the same principle in control theory so that every subsystem can gather information from the environment and make decisions based on modeling, planning, and optimization. For each subsystem, there is a local feedback loop and a knowledge base, which plays a central role in the control loop. Fig. 2 shows a typical structure of

an autonomic defense subsystem. In Fig.2, the autonomic defense subsystem has a sensor and an effector for gathering data and controlling the protected information system, respectively. The data from the sensor are processed by different information processing modules including monitoring, analysis, planning and execution. During the processing of data and information, a knowledge base interacts with each module to derive, utilize and modify the knowledge about the protected information system and its external environment. In addition, the autonomic defense or security subsystem also has external sensors and effectors from other autonomic subsystems so that it may also be a protected system of other defense systems. Based on the above architecture, the autonomic DDIAS can be designed to have the ability of self-management, which can realize self-configuration, self-healing, self-optimization, and self-protection. As a result, the survivability of information systems can be greatly improved in dynamic environment.

4 Autonomic Distributed Incident Recovery as a Case Study

In this section, we will present a case study of autonomic computing for DDIAS. The case study is concerned with the last defense layer of DDIAS, i.e., emergency response and incident recovery. The research work towards autonomic distributed incident recovery was recently studied in [10], where the tenure duty method (TDM) was proposed for dynamic service backup and roaming. In [10], information service is provided by a set of servers called backup pool and only one server can provide service outside. Multiple servers are cooperated to improve the survivability of information service systems. The service time of a server, from its beginning to translating service to another server, is called tenure. The basic idea of TDM is to make every server be fully responsible for the information service task during its duty period and the alternation among the server agents are carried out by a tenure time scheduling mechanism so that autonomic service roaming can be realized to improve survivability under different attacks. In [10], a random tenure rule is proposed to make the alternation of servers unpredictable. In this paper, we propose to use a combination of random TDM and adaptive TDM based on information feedback so that autonomic distributed incident recovery of information service can be realized in an efficient way.

In adaptive TDM, information feedback is from outside intrusions. An adaptive tenure time scheduling mechanism is designed according to the observed intrusion types, i.e., the tenure time of every server is adjusted with the variation of intrusion types so that the survivability of information service can be improved. The idea behind this adaptive TDM is that different servers may have different immune ability for attacks and the system security can be improved when a server with immune ability prolongs its tenure time. Let G denote a server, s denote the mean tenure time of G and I denote the intrusion set. Let I_G denote the set of intrusion types that server G will be attacked and stop service. The detailed description of the adaptive TDM combined with random TDM is presented as follows.

(Random and Adaptive TDM)

Randomly generate tenure time s_i using a mean value s for each server.

If no attacks happen during the tenure time of G

$s \leftarrow s + \Delta t$

Else

If attack i happens,

If the attack fails, and $i \in I_G$ // G has immune ability for i

$I_G \leftarrow I_G - \{i\}$

End if // Remove i from the intrusion set of G

If the attack succeeds

If $s > \Delta t$ $s \leftarrow s - \Delta t$ End if // Decrease mean tenure time

If $i \notin I_G$

$I_G \leftarrow I_G \cup \{i\}$ // Add i to the intrusion set of G

End if

End if

$I = \bigcup_G I_G$ // Recalculation of total attack set

End if

End if

Based on the random and adaptive TDM, each server runs the following process to realize autonomic service roaming.

(Running Process of Servers)

Begin running normal service process, generate random tenure time s_i , start counting time, $t=0$.

A: Run normal service, counting time t

If $t=s_i$ and no attacks,

Start updating log and synchronize backup information

$s=s+\Delta t$

Stop normal service, change to backup state, exit.

Else if attacks happen,

Using random and adaptive TDM to update tenure time

If service is recoverable, recovery, return to **A**

Else stop service, exit.

To illustrate the effectiveness of the autonomic distributed service roaming for incident recovery, we conduct experiments on continual HTTP service using three backup servers in a network environment. In the experimental setup, a client visits Web service through the network and three servers constitute a service backup pool. At anytime, the client only visits one of the three nodes and the service is alternately provided by the three servers. The tenure time of each server is determined by the random and adaptive TDM presented above. When a server stops its service, the other two servers will compete for the next service period.

The attack set is simulated by detection time T_d , i.e., different T_d corresponds to different attacks. The shorter is the detection time, the higher will be the attacker's ability to destroy the service system. The risk level is simulated both by the detection time and the interval T_u between attacks. The destroy rate ρ (rou) or the success rate of attacks is computed to evaluate the survivability of the system. The experimental results are shown in Fig.3.

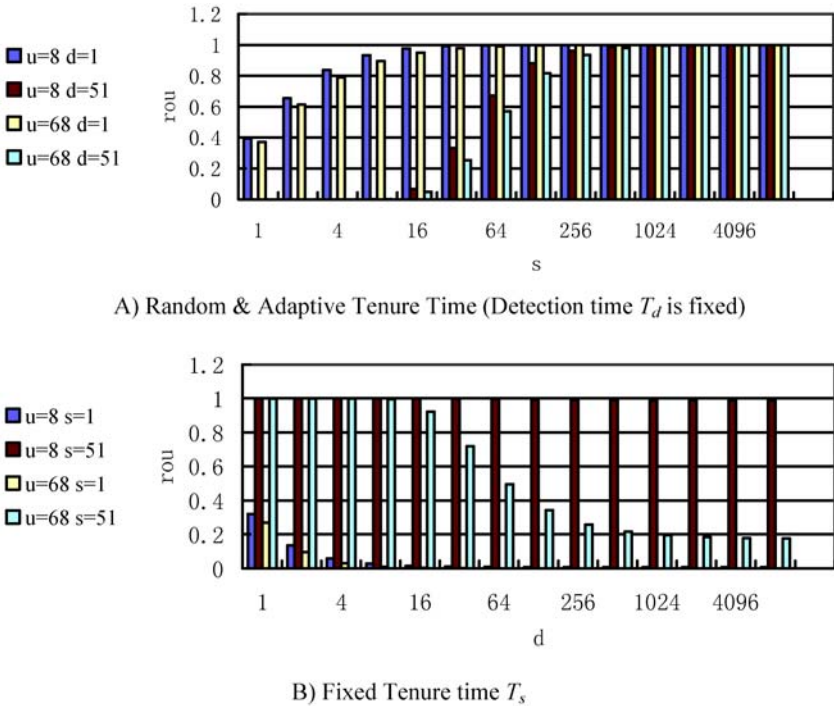


Fig. 3. The relationship between tenure time, risk level and attack success rate

Fig.3 A) shows the variations of attack success rate ρ (rou) under random and adaptive tenure time, attack interval u and detection time d . Fig.3 B) shows the result when the tenure time of each server is fixed. Note that the two figures do not have the same axes. From the above results, it can be inferred that when the tenure time is randomly determined by the random and adaptive TDM, the attack success rates are much lower than those under fixed tenure time in Fig.3 B). For example, when $u=8$ and $s=51$, the attack success rates in Fig.3 B) are almost equal to 1 for any detection time d . While in Fig.3 A), when $u=8$, $d=51$, the attack success rates can also be relatively low even when $s=64$. Thus, it can be concluded that the autonomous service roaming based on random and adaptive TDM can improve the survivability of information system without sacrificing much service quality since too short tenure time will cause frequent alternation of service, which will degrade the service quality.

5 Conclusion and Future Work

This paper proposes an autonomic computing architecture for defense-in-depth information assurance systems. The autonomic computing architecture uses multi-loops of information feedback to realize self-management of DDIAS so that the increasing complexity of DDIAS can be efficiently managed by autonomous subsystems. A case study of autonomic service roaming for incident recovery is presented and experimental results illustrate the effectiveness of the proposed method. Future work may include the design and implementation of autonomic computing mechanisms for different defense subsystems as well as the whole DDIAS using risk evaluation as a main feedback loop.

References

1. McHugh, J., Christie, A., Allen, J.: Defending Yourself: The Role of Intrusion Detection Systems. *IEEE Software* (2000) 42-51
2. Huang, Z.G., Lu, X.C., Wang, H.M.: A Diversified Dynamic Redundancy Method Exploiting the Intrusion Tolerance. In: *Proceedings of the Third Information Survivability Workshop (ISW-2000)*, October 24-26, Boston MA, USA (2000)
3. Deep, M., David T.: Multi-Layered Network Survivability – Models, Analysis, Architecture, Framework and Implementation: An Overview. In: *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX' 2000)*, Hilton Head Island, South Carolina, January 25-27 (2000)
4. Information Assurance through Defense-in-Depth. Directorate for Command, Control, Communications, and Computer Systems. U.S. Department of Defense Joint Staff, February (2000)
5. Ganek, A. G., et al.: The Dawning of the Autonomic Computing Era. *IBM Systems Journal* 42 (2003) 5-18
6. Haas, R., Droz, P., Stiller, B.: Autonomic Service Deployment in Networks. *IBM Systems Journal* 42 (2003) 150-164
7. Bantz, D. F., Bisdikian, C., et al.: Autonomic Personal Computing. *IBM Systems Journal* 42 (2003) 165-176
8. Horn, P.: Autonomic Computing: IBM's Perspective on the State of Information Technology, IBM Corporation. available at http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf. (2001)
9. Bass, T., Robichaux, R.: Defense-In-Depth Revisited: Qualitative Risk Analysis Methodology for Complex Network-Centric Operations. In: *Proc. of IEEE MILCOM 2001*, October 28-31 (2001) 64-70
10. Huang, Z.G.: The Tenure Duty Method (TDM) in the Active Incident Recovery Research. *Lecture Notes in Computer Science* 2834 (2003) 557-564

Mining Maximal Frequent Itemsets for Intrusion Detection*

Hui Wang^{1,2}, Qing-Hua Li², Huanyu Xiong¹, and Sheng-Yi Jiang²

¹ Wuhan Communication College, 430010, Wuhan, China

² Computer School, Huazhong University of Science & Technology, 430074, WuHan, China
hustwanghui@mail.china.com

Abstract. It has been the recent research focus and trend to apply data mining techniques in an intrusion detection system for discovering new types of attacks, but it is still in its infancy. This paper presents an innovative technique, called *MMID*, that applies maximal frequent itemsets mining to intrusion detection and can significantly improve the accuracy and performance of an intrusion detection system. The experimental results show that *MMID* is efficient and accurate for the attacks that occur intensively in a short period of time.

Keywords: data mining, intrusion detection, maximal frequent itemset

1 Introduction

In today's information age, where nearly every organization is dependent on the Internet to survive, it is imperative to guarantee the security of computer systems. As a powerful weapon to protect networks, intrusion detection system (IDS) has gained more and more attention. It is maintained by monitoring audit trail data. Currently, most IDSs are developed by manual and ad hoc means. With the amount of information passed over networks and the very size of these networks has been increasing exponentially, the so-called expert knowledge is often limited and unreliable. On the other hand, data mining approaches can be used to find unknown patterns hidden in the vast amount of audit trail data so that it can be more objective than the ones hand-picked by experts. Therefore, data mining approaches can be very promising for intrusion detection system development. This is just one of the motivations to apply data mining to intrusion detection.

The intrusion detection systems are based on the belief that an intruder's behavior will be noticeable different from that of a legitimate user and that many unauthorized actions are detectable. They collect and monitor operating system and network activ-

* This paper is supported by the National Natural Science Foundation of China under Grant No.60273075.

Hui Wang is a professor in Wuhan Communication College, and a PhD candidate in computer school, Huazhong Univ. of Sci. & Tech., China. Her research interests include intrusion detection and data mining.

ity data, and analyze the information to determine whether there is an attack occurring. There are two major categories of analysis: misuse detection and anomaly detection. Misuse detection uses the signatures of known attacks to identify a matched activity as an attack instance, while anomaly detection uses established normal profiles to identify any unacceptable deviation as the result of an attack. Usually, misuse detection is more effective against known attacks with higher true positive rate, while anomaly detection could catch new attacks but with higher false positive rate.

This paper presents an innovative technique, called *MMID* (Mining Maximal for Intrusion Detection), that applies maximal frequent itemsets mining to intrusion detection and can significantly improve the accuracy and performance of an intrusion detection system. The experimental results show that *MMID* is efficient and accurate for the attacks that occur intensively in a short period of time.

The rest of this paper is organized as follows: We start from looking at related work in section 2. In section 3 we present *MMID*. We give experimental results in section 4 and conclusions in section 5.

2 Related Work

Although intrusion detection techniques have been studied for more than two decades, they are still at the fairly primitive stage. STAT[1], IDIOT[2] and NIDES[3] are influential.

It has been the recent research focus and trend to apply data mining techniques in an intrusion detection system for discovering new types of attacks, but it is still in its infancy. Warrender et al.[4] showed that a number of machine-learning approaches, e.g., rule induction, can be used to learn the normal execution profile of a program, which is the short sequences of its run-time system calls made. Lee[5] used a machine learning classifier, RIPPER, to produce rules to classify system call sequences as “normal” or “abnormal”. Karlton Sequeira and Mohammed J. Zaki[6] worked with user command-level data to recognize masquerader. Daniel Barbará et al.[7] developed a test bed for exploring the use of data mining in intrusion detection.

Association rules which come from frequent itemsets mining are used in many intrusion detection systems[5][7] but showed poor performance and limited accuracy. Factually, frequent itemsets, more directly, maximal frequent itemsets, can be used to detect intrusions, in such way, needless learning association rules from frequent itemsets and the performance of intrusion detection system can be improved. In section 3 we describe such new way in more detail.

3 MMID

MMID is a network anomaly detection system that works on tcpdump data, which based on schema $R: R(Ts, Src.IP, Src.Port, Dst.IP, Dst.Port, Serv, label)$. Where, Ts means the beginning time of a connection, $Src.IP$ and $Src.Port$ refer to source IP and port number respectively, and $Dst.IP$ and $Dst.Port$ represent destination IP and port

number, Serv means service type, label represents the connection record is normal or intrusive or suspicious or infrequent.

MMID is based on the two assumptions:

Assumption 1: The activity that occurs frequently in attack free training data set is normal behavior of the system. So we use the set of the maximal frequent itemsets over the attack free training data set to be the profile of the system and users' normal behavior.

Assumption 2: The activity that occurs frequently in a short period of time and can't be covered by the system's profile is abnormal. Therefore we use the set of the maximal frequent itemsets over the training data set with attacks in it to be the model of attacks.

There are two main stages in our approach to mining intrusions. In the training phase the system and user profiles, as well as the attack models, are created, and in the testing phase the connection record in a sliding window is to be labeled as *intrusive*, *suspicious*, or *normal* against the corresponding profiles and attack models. The complete architecture of MMID is shown in Fig. 1. The normal profiles and the attack models are updated by those suspicious maximal frequent itemsets under the help of human experts.

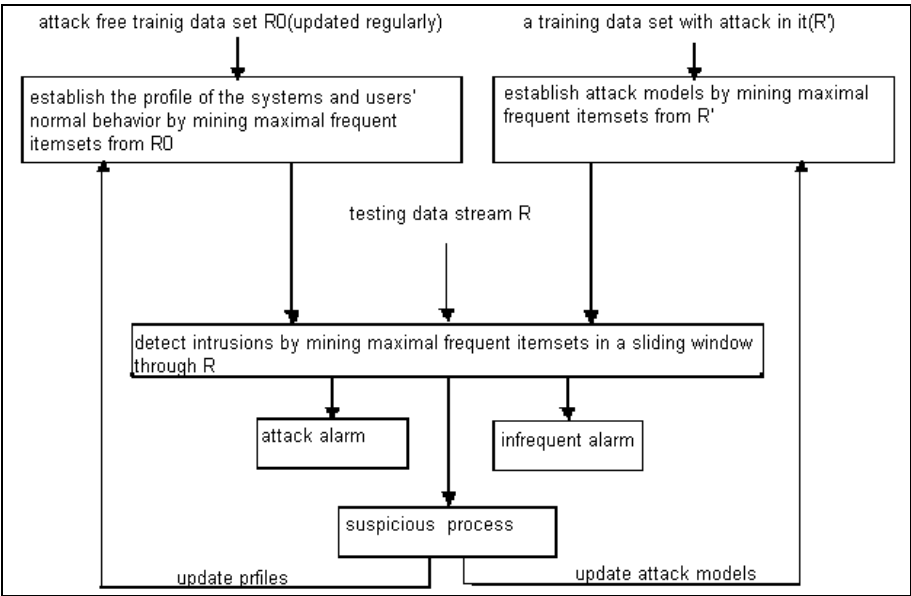


Fig. 1. The architecture of MMID

We have several definitions to describe MMID as follows:

Definition 1: priority of the labels

Priority(*intrusive*)>priority(*suspicious*)>priority(*normal*)>priority(*infrequent*)

Definition 2: labeling function

Let record r has the existing label *oldlabel*, *newlabel* be a new label. Define label function *assignlabel()* as follows:

$$\text{assignlabel}(r, \text{oldlabel}, \text{newlabel}) = \begin{cases} \text{newlabel} & \text{if } \text{priority}(\text{newlabel}) > \text{priority}(\text{oldlabel}) \\ \text{oldlabel} & \text{otherwise} \end{cases}$$

Table 1 gives the related attributes and interesting patterns for intrusion detection. Table 2 shows tokens that would be used for describe MMID. MMID detects intrusions by labeling each link record in sliding window that was shown in Algorithm 1.

Table 1. The related attributes and the interesting patterns used by MMID

Attributes		Patterns	
A	<i>source_ip</i>	P ₁	{C}
B	<i>source_port</i>	P ₂	{AC}
C	<i>destination_ip</i>	P ₃	{ADE}
D	<i>destination_port</i>	P ₄	{ABC}
E	<i>Service type</i>	P ₅	{ABDE}
Ts	<i>start_time</i>	P ₆	{ACDE}
		P ₇	{ABCDE}

Table 2. The tokens used by MMID

Tokens	Token's Meaning
R_0	Attack free training data for establishing the profile of the system and users' normal behavior.
R'	Training data with attack in it for establishing the models of attacks.
m	The set of the maximal frequent itemsets over R_0 which represents the profile of the system and users' normal behavior.
$\hat{m}$	The set of the maximal frequent itemsets in sliding windows over R' which represents the attack models.
$\tilde{m}$	The set of the maximal frequent itemsets in the current sliding windows.
$w[t_1, t_2]$	A sliding window which includes all the connection records started in the time between $[t_1, t_2]$.
l	The length of sliding window.
h	The sliding step of sliding window.
m_i	The set of frequent itemsets corresponding to the pattern $P_i (1 \leq i \leq 7)$
r_i	The i^{th} pattern for record $r. (1 \leq i \leq 7)$

Algorithm 1: MMID**Input:** R, ε_2, l

```

/* MMID labels for each connection record in  $R$  */
1.  MinMax_for_IDS( $w[t_1, t_2], \varepsilon_2, \tilde{m}$ ) /*scan the sliding
window to count  $\tilde{m}$  */
2.  /* label each connection record in the current
sliding window */
3.  for  $\forall r: r \in w[t_1, t_2]$ 
4.    for  $i:=1$  to 7 do
5.      if  $\exists \tilde{p}: \tilde{p} \in \tilde{m} \wedge \tilde{p} \supseteq r_i$  then
6.        if  $\exists \hat{p}: \hat{p} \in \hat{m} \wedge \hat{p} \supseteq r_i$ 
7.          then assignlabel( $r, oldlabel, intrusive$ )
8.        else
9.          if  $\exists p: p \in m \wedge p \supseteq r_i$ 
10.           then assignlabel( $r, oldlabel, normal$ )
11.         else assignlabel( $r, oldlabel, suspicious$ )
12.       endif
13.     endif
14.   else assignlabel( $r, oldlabel, infrequent$ )
15.   endif
16. endfor
17. endfor
18. delete all records with label intrusive from  $w$ 
19. /* slide the window */
20.  $t_1' :=$  the start time of the first record left in the
window
21.  $t_2' := t_1' + l - 1$ 
22. while window  $[t_1', t_2']$  includes no new records do
23.    $t_1' := t_1' + 1$ 
24.    $t_2' := t_1' + l - 1$ 
25. endwhile
26.  $t_1 := t_1'$ 
27.  $t_2 := t_2'$ 
28. goto 1
29. end

```

4 Experiment Results

In this section, we report our experimental results on training and testing datasets for three weeks at (http://www.ll.mit.edu/ist/ideval/dat/1998/1998_data_index.html). All the experiments are performed on a 1.7GHz Pentium PC machine with 256 MB main memory, running on Microsoft Window XP. All the programs are written in Microsoft Visual C++6.0. We group the datasets into 5 different sets of A, B, C, D and E, where all the records in A are labeled with *normal*, records in B, C, D and E are labeled with *normal* or *intrusive*. B and C come from the same network with A, while D and E come from a network different from A. Let D_0 represent the subset of D where all the records in D_0 are labeled with *normal*, D_1 labeled with *intrusive*.

Let tp represent true positive rate, fn represent false negative rate, fp represent false positive rate, tn represent true negative rate, sp represent suspicious positive rate and sn represent suspicious negative rate. They are computed as follows.

$$tp = \frac{\text{the number of intrusive records labeled with intrusive by MMID}}{\text{the total number of intrusive records in dataset}} * 100\%$$

$$fn = \frac{\text{the number of intrusive records labeled with normal by MMID}}{\text{the total number of intrusive records in dataset}} * 100\%$$

$$fp = \frac{\text{the number of normal records labeled with intrusive by MMID}}{\text{the total number of normal records in dataset}} * 100\%$$

$$tn = \frac{\text{the number of normal records labeled with normal by MMID}}{\text{the total number of normal records in dataset}} * 100\%$$

$$sp = \frac{\text{the number of intrusive records labeled with suspicious by MMID}}{\text{the total number of intrusive records in dataset}} * 100\%$$

$$sn = \frac{\text{the number of normal records labeled with suspicious by MMID}}{\text{the total number of normal records in dataset}} * 100\%$$

Giving different minimum relative support threshold ϵ_1 for normal frequent behavior, ϵ_2 for abnormal frequent behavior, window length l and sliding step h , MMID are tested over different datasets of training and testing data. Table 3~6 show the results under the condition of $\epsilon_1=0.5\%$ and $h=2(\text{secs})$.

In experiment group 1, A and B are training data, C is the testing data. The measures on the detection accuracy vary with the window length, sliding step and support threshold. We noticed that the suspicious records are not many. This is because C is from the same network with A and B, the profile can basically cover C.

In experiment group 2, A and B are keeping as training data, D is the testing data. Lots of records are labeled with *suspicious* by MMID. It is reasonable because D is from a different network from A and B, the profile established on A and B can't cover D well.

In experiment group 3, D_0 and A and B are training data, D is keeping as testing data. The experiment results show that the number of normal records labeled as *suspicious* by MMID is much less than group 2, while the number of intrusive records labeled as *suspicious* by MMID is keeping high. This is because that D_1 is not training data, so the attack models can't recognize the intrusive records well.

In experiment group 4, A, B and D are training data, E is testing data, the results are similar with group 1. It shows that MMID could work well with the profiles and attack models are updated.

Table 3. The Performance of MMID (Group 1)

No.	$\varepsilon 2$	l	sp	tp	fn	sn	fp	tn
1	80	3	1.23	90.50	7.30	3.18	5.39	84.60
2	80	5	1.23	90.03	7.62	3.20	5.42	84.65
3	50	3	1.76	90.30	6.76	5.72	7.32	82.01
4	50	5	1.81	90.01	6.96	5.72	7.03	81.75
5	90	3	0.64	91.05	8.05	1.68	3.15	87.75
6	90	5	0.57	90.81	8.30	1.75	3.24	87.63

Table 4. The Performance of MMID (Group 2)

No.	$\varepsilon 2$	l	sp	tp	fn	sn	fp	tn
1	80	3	95.60	0	0	82.30	0	0
2	80	5	95.73	0	0	82.25	0	0
3	50	3	95.40	0	0	83.11	0	0
4	50	5	95.51	0	0	83.85	0	0
5	90	3	95.82	0	0	88.05	0	0
6	90	5	95.81	0	0	88.13	0	0

Table 5. The Performance of MMID (Group 3)

No.	$\varepsilon 2$	l	sp	tp	fn	sn	fp	tn
1	80	3	95.60	0	0	3.58	4.34	81.67
2	80	5	95.73	0	0	3.60	4.42	81.65
3	50	3	95.40	0	0	5.82	6.62	81.01
4	50	5	95.51	0	0	5.42	6.43	81.15
5	90	3	95.82	0	0	1.98	3.65	86.15
6	90	5	95.81	0	0	1.65	3.64	84.53

Table 6. The Performance of MMID (Group 4)

No.	$\varepsilon 2$	l	sp	tp	fn	sn	fp	tn
1	80	3	1.13	90.30	7.10	3.38	5.34	84.60
2	80	5	1.14	90.33	7.12	3.30	5.32	84.65
3	50	3	1.67	90.10	6.26	5.42	7.92	82.01
4	50	5	1.57	90.11	6.56	5.52	7.43	81.75
5	90	3	0.90	91.02	8.01	1.78	3.45	87.75
6	90	5	0.98	90.81	8.01	1.75	3.64	87.63

5 Conclusions

We have proposed an innovative technique *MMID* in this paper that applies maximal frequent itemsets mining to intrusion detection and can significantly improve the accuracy and performance of an intrusion detection system. The experimental results show that *MMID* is efficient and accurate for the attacks that occur intensively in a short period of time.

References

1. Porras P. A. and Kemmerer R. A. Penetration state transition analysis: A rule-based intrusion detection approach. In: Proceedings of the Eighth Annual Computer Security Application Conference. San Antonio TX. 1992. 220-229
2. S. Kumar and E.H. Spafford. A software architecture to support misuse intrusion detection. In: Proceedings of the 18th national information security conference. 1995. 194-204
3. Anderson D., Teresa F. Lunt, Javitz H. et al. Detecting Unusual Program Behavior Using the Statistical Component of the Next-generation Intrusion Detection Expert System (NIDES). SRI-CSL-95-06, SRI International Menlo Park CA, 1995.
4. C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, May 1999.
5. Wenke Lee. A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems:[PhD thesis]. Columbia University, 1999.
6. Karlton Sequeira, Mohammed J. Zaki. ADMIT: Anomaly-based Data Mining for Intrusions. in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton Alberta Canada. 2002. 386-395
7. Daniel Barbará, Julia Couto, Sushil Jajodia, et al. ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection. SIGMOD, 2001, 30(4): 15-24

Context-Aware Role-Based Access Control Model for Web Services

Xu Feng, Xie Jun, Huang Hao, and Xie Li

Department of Computer Science and Technology
Nanjing University, Nanjing 210093, China
njxuf@163.com

Abstract. This paper first brief reviews the state of the security technology research and access control in the web service environment. Despite recent advances in access control approaches for Web Services, the heterogeneity of subjects and objects in Web Service environment has made it difficult to development of effective access control system. So we present a context-aware service-orient role-based access control model (CSRBAC). In CSRBAC model, access control system can make its access control decisions by capturing security relevant environmental context, such as time, location, operation state, or other environmental information. Based on CSRBAC model, a secure architecture model for Web Services is presented. It implements an access control system with dynamically grant and adapt permissions to users based on their current context. Compared to traditional access control mechanisms, the CSRBAC model can provide management flexibility and improved security for Web Services applications.

1 Introduction

Web Services provide a new technology to construct dynamic computing platform. This paradigm has been applied in various fields such as electronic shopping, trading with information, applications in the telemetric area, or even accessing grid computing services via the web. At present, along with Web Services technology application and the promotion, the security problem has already become the key factor which restricts it further to develop.

The access control mechanisms required by distributed, heterogeneous domains or systems are becoming increasingly complex. The complexity arises not only caused by the huge number of the distributed clientele accessing online services but also heterogeneity of subjects and objects. The heterogeneity means that the user profile may change dynamically, and hence access control system should make its access control decisions by capturing security-relevant environmental context, such as time, location, operation state, or other environmental information available when the access requests are made.

The characteristic of Web Services requires an access control model that offers specific capabilities. In this paper a role-based access control model is presented, to address a new set of challenges that traditional security models do not address.

The remainder of this paper is organized as follows: The second section brief discusses the related work and presents the motivation of the paper. The third section presents a context-aware service-oriented role-based access control model (CSRBAC model). The section 4 presents a secure architecture model for Web Services based on CSRBAC model. In the section 5 the conclusion is given and the problems are pointed out, which should be resolved in further research.

2 Related Works and Motivation

Access control for Web Services is already becoming the hot topic in the field of Web Services security. Several correlated specifications [4]-[7] are proposed toward providing a comprehensive standards framework for secure Web Services applications.

SAML [4] is an XML-based framework for request/response exchanges of authentication and authorization information. XACML [5] is an XML specification for expressing fine-grained information access policies in XML documents or any other electronic resource. XrML [6] is a general-purpose, XML-based specification for expressing rights and conditions, such as expiration times, associated with digital resources and services. XrML focuses on digital rights management, but it overlaps with XACML. XACML is the more comprehensive and flexible specification.

Moreover some specifications [7-9] have studied the security of SOAP messages. Among these WS-Security [7] is one of the most representations. WS-Security describes enhancements to SOAP messaging to provide protection through message integrity, message confidentiality, and single message authentication. But WS-Security does not ensure security nor does it provide a complete security solution.

The RBAC model is widely accepted recently. Considered that XACML does not directly support the notion of roles, Bhatti [10] proposes an XML-based RBAC policy specification framework for enforcing access control on XML documents. But control access for web service isn't taken into account. So we present a service-oriented role-based access control model [14]. It is suitable for the characters of service-oriented architecture of the Web Services.

Because of the characteristic of Web Services and the complexity of the distributed environment, its security is big challenge problem. In [13] we discuss the characters of these above works and point out the questions should be resolved. Amongst existing models [10-12, 14] there are the lacks of context-aware models for Web Services access control. We next elaborate on these issues, and propose a context-aware service-oriented role-based access control model in an attempt to address them.

3 CSRBAC Model

In RBAC [1-3] model, permissions are associated with roles, and users are made members of appropriate roles. This greatly simplifies management of permissions. Roles are closely related to the concept of user groups in access control. However, a role brings together a set of users on one side and a set of permissions on the other,

whereas user groups are typically defined as a set of users only. RBAC [1-3] is a promising alternative to traditional discretionary and mandatory access controls, and ensures that only authorized users are given access to certain data or resources.

Although RBAC has so many advantages, it can not completely suit for Web Services environment.

3.1 Basic Ideal

The architecture of Web Services is service-oriented, and Web Services need to cooperate with each other across heterogeneous domains. So we propose a context-aware service-oriented role based access control (CSRBAC) model. It is shown in Figure 1. In CSRBAC model, traditional protected objects and operations are replaced by services.

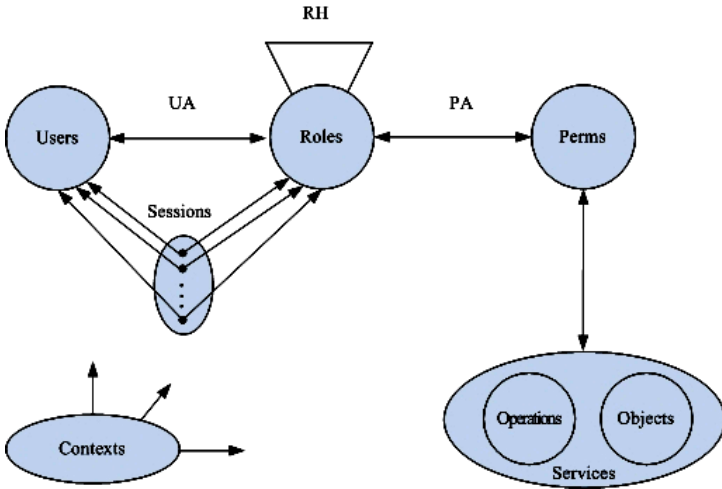


Fig. 1. CSRBAC model

The following definition formalizes the CSRBAC model.

Definition 1: (CSRBAC model): CSRBAC model is composed by the below entity set and the relations:

- U, R, P, S, C , and Srv (users, roles, permissions, sessions, contexts, and services respectively),
- $PA, PA \subseteq R \times P$, a many-to-many permission to role assignment relation,
- $UA, UA \subseteq U \times R$, a many-to-many user to role assignment relation,
- $user : A \rightarrow U$, a function mapping each session s_i to the single user $user(a_i)$,
- $roles : A \rightarrow 2^R$, a function mapping each session s_i to a set of roles,
- Srv : a set of services,
- RH (role hierarchy), $RH \subseteq R \times R$, a many-to-many role to role relation.

In CSRBAC model, a service is composed of several operations on objects. A service is an abstraction of the operations provided by the system on its objects. Contexts represent the sets of security-relevant context information in the system, e.g. identification, time, location, operation state, or other environmental information.

In order to formalize the service and context, we introduce two items to allow specifying domains of legal values for various context parameters. Our formal model relies on the items we define below:

Atom Service: is a service which can't be divided. It is represented by $asrv$. It can be defined as a tuple $\langle operation, object \rangle$.

It is used to denote the possible component of a service.

Context Parameter: is represented by a parameter expression $cont$.

Definition 2: (Service): A service set Srv may consists of n atom services $\{asrv_1, \dots, asrv_n\}$.

Definition 3: (Context): A context set C may consists of n context parameters expression $\{cont_1, \dots, cont_n\}$, $n \geq 0$, for any $cont_i$ and $cont_j$ $cont_i \neq cont_j$, with $i \neq j$ and $1 \leq i, j \leq n$, we have that $cont_i \neq cont_j$, (i.e. the parameter expression must be distinct).

The CSRBAC model also supports three well-known security policies: data abstraction, least-privilege assignment, and separation of duties.

3.2 Role Assignment and Activation

In traditional RBAC model, system administrators can create predefined roles, grant permissions to those roles, and then assign users to the roles on the basis of their special job responsibilities and system security policy. Therefore, role-permission relationships can be predefined, which makes it simple to assign users to the predefined roles. In general, when a user access system resource, if he is assigned to several roles, it is up to him to decide which role(s) he is authorized to activate.

In CSRBAC model, the user does not select the role to be activated directly. Instead, the role activation depends on the security-relevant environmental context. This means that dependent on context, the user role(s) to be activated is selected by the access control system.

When the online services system receives a request, roles will be activated dependent on whether he has already registered, where he send the request, security token which he presents and for which services he subscribed at registration. Actual roles will be activated based on the fulfillment of these requirements. The procedure deciding which roles are selected for activation is depicted in Figure 2. In the figure, a simple example is denoted.

In order to describe how and which roles are activated we introduce some items.

Given a user u , $r(u)$ is denote to the set of roles for which is predefined to assign to the user u . so we can get:

$$r(u) = \{r \in R, | (u, r) \in UA\}$$

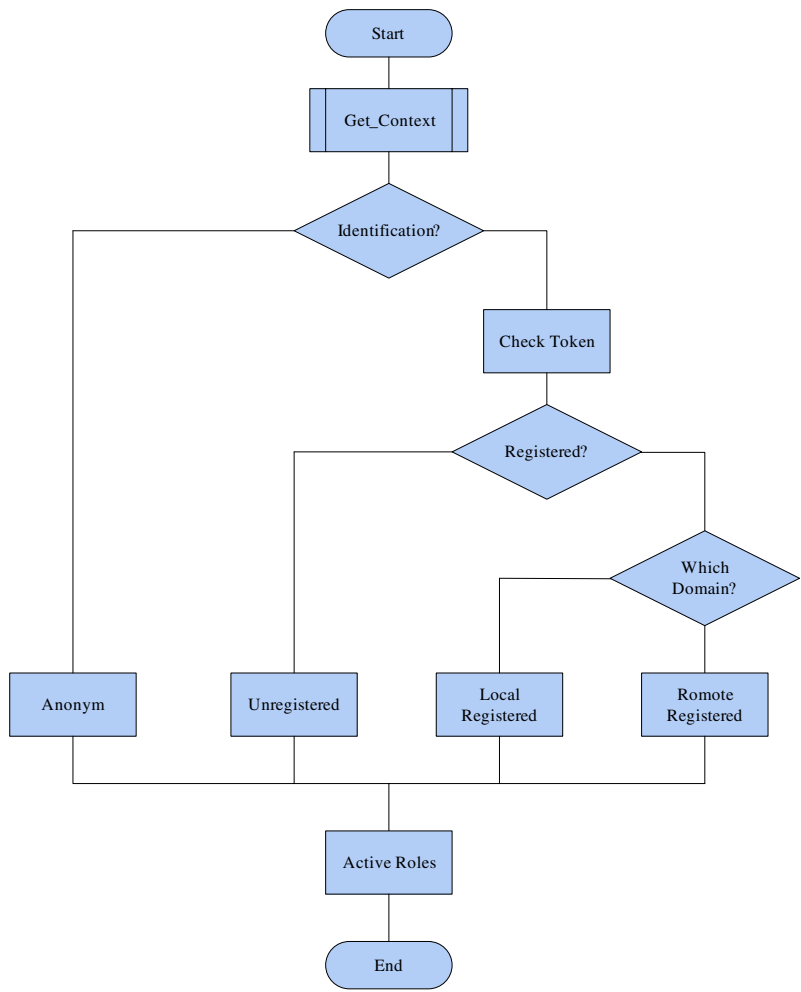


Fig. 2. The Procedure of role activation

For formalization reasons, we give following define.

Definition 4: (Activated role): Given $cont_i \in C$ ($1 \leq i \leq n$, n is integer), we can get:

$$r(u) \wedge cont_1 \wedge \dots \wedge cont_n \rightarrow May\_activated(u, r)$$

It is obvious that $May\_activated(u, r)$ is subset of $r(u)$.

There are two ways to deciding how to activate role for a user. One is the method to decide which roles may be activates according to the actual context information. This means that in order to realize access control, system should decide which role to be activated actually depended on predefined roles assigned to the user and context condition. Another is the way to partition predefined role set according to possible context information. In other words when system defines user assignment, roles classification will be made based on important context parameter requirement. When role

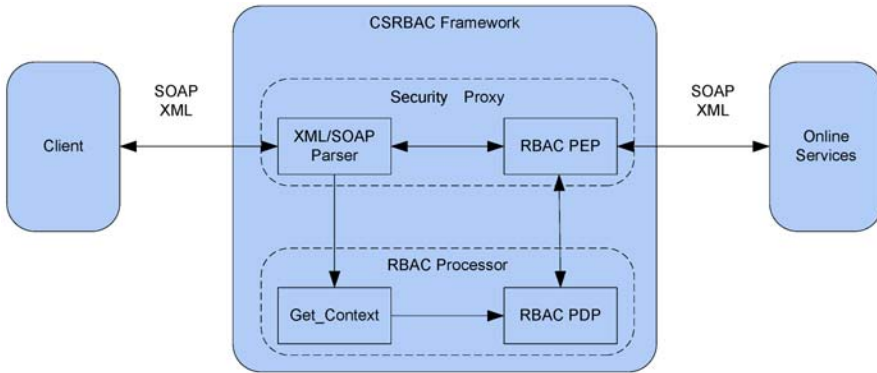


Fig. 3. Security architecture for Web Services

activation system only need to select roles from the predefined role sets depend on the context condition and user identification.

But for most practical purposes, the set context parameters should be extended according to the system requirements in order to define access conditions based on appropriate security context.

4 Implementation Architecture

Based on CSRBAC model, we design a security architecture for Web Services which is shown in Figure 3. In this CSRBAC framework, client end is service requestor, soap proxy and Web Services are service providers.

As indicated in the figure, the two main subsystem of CSRBAC Framework are Security Proxy and RBAC Processor. Security Proxy contains XML/SOAP Parser (a parser for XML/SOAP protocol) and RBAC PEP (a RBAC policy enforcement Point Module). RBAC Processor administers and makes decision according to defined policies. RBAC Processor contains two subcomponents, Get_Context module and RBAC PDP (RBAC policy decision point) module.

In CSRBAC Framework, security proxy parser the information from client, and then forwards client request on to RBAC Processor, where authentication and role assignment are processed. RBAC Processor may contain authentication module, or pass the identifier information to third party certificate authority. Get_Context module extracts the identifier information and context from the client request, and then sends it to RBAC PDP. Finally according to the role of requestor, the request will be rejected or accept by security proxy. So according to authentication and role assignment, users can obtain the authorized service.

5 Conclusion

In this paper, we present a CSRBAC model for access control in Web Services. It supports context-aware access control based on security context condition. The secu-

urity context may be time, place, and identification and so on. Our CSRBAC model can simplify the role assignment and management in heterogeneous system. Compared to traditional access control mechanisms, the CSRBAC model can provide management flexibility and improved security for Web Services applications.

References

1. Ravi Sandhu, Edward Coyne, Hal Feinstein, Charles Youman. Role-Based Access Control Models IEEE Computer, Volume 29, Number 2, pages 38-47, February 1996.
2. D. F. Ferraiolo , R. Sandhu , S. Gavrila , D. R. Kuhn ,R. Chandramouli, "Proposed NIST standard for role-based access control", ACM Transactions on Information and System Security (TISSEC), Volume 4, Issue 3, August 2001.
3. J Park, R Sandhu. Secure Cookies on the Web. IEEE Internet Computing, 2000, 4 (4): 36-45.
4. OASIS Standard. Security Assertion Markup Language, SAML1.1, Oct.2003.
<http://www.oasis-open.org/committees/security/docs/cs-sstc-core-01.pdf>.
5. OASIS Standard. XACML 1.0 Specification Set. Feb. 2003.
<http://www.oasis-open.org/committees/xacml/>.
6. ContentGuard, Inc. eXtensible Rights Markup Language, XrML 2.0. (2001).
<http://www.xrml.org>.
7. Web Services Security (WS-Security) version1.005, April 2002.
<http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>
8. Web Services Security Core Specification Working Draft 01, 20 September 2002.
<http://lists.oasis-open.org/archives/wss/200209/pdf00000.pdf>
9. W3C NOTE. SOAP Security Extensions: Digital Signature.
<http://www.w3.org/TR/SOAP-dsig>.
10. Rafae Bhatti, James B. D. Joshi, Elisa Bertino, Arif Ghafoor. Access Control in Dynamic XML-based Web-Services with X-RBAC", Accepted for publication in The First International Conference on Web Services, Las Vegas, June 23-26, 2003.
11. Ernesto Damiani , Sabrina de Capitani di Vimercati , Stefano Paraboschi , Pierangela Samarati, Design and implementation of an access control processor for XML documents, Computer Networks: The International Journal of Computer and Telecommunications Networking, v.33 n.1-6, p.59-75, June 2000.
12. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati. A Fine-Grained Access Control System for XML Documents, in ACM Transactions on Information and System Security (TISSEC), vol. 5, n. 2, May 2002, pp. 169-202.
13. Xu Feng, Lin Guoyuan, Huang Hao. A Survey on Access Control for Web Services. Accepted for publication in Computer Science (in Chinese).
14. Xu Feng, Lin Guoyuan, Huang Hao, Xie Li. Role-Based Access Control System for Web Services. Accepted for publication in The 2004 International Conference on Computer and Information Technology, Wuhan, China, September 14-16, 2004.

Policy-Tree Based Proactive Defense Model for Network Security

Feng Zhang¹, Zhiguang Qin², and Shijie Zhou²

¹ ZTE Corp, Gao Peng Road 3, 610041 Chengdu, P.R. China
zhangfeng@uestc.edu.cn

² School of Computer Science and Engineering, UESTC, 610054 Chengdu, P.R. China
{qinzg, sjzhou}@uestc.edu.cn

Abstract. In-depth defense for network security offers promotion on robusticity and survivability of information system. It prevents attacker from damaging system even he has already broken through one or several but not all layers of the system. Proactive defense integrates in-depth defense and shows the activeness greatly in contrast with traditional defense. It predicts intrusion trend and obtains attacker's information, dynamically evaluates and responds to intrusion. This reflects the counteracting property of security. Formally defined in Z language, policy-tree model for proactive defense is proposed in this paper. Moreover, completeness, correctness and consistency are analyzed. A completely building method, an abstract for correctness validating and an auto consistency checking method on security policy are designed. Policy-tree model gives theoretical and methodological support for proactive defense.

1 Introduction

In information era, requirement for Security is changing from information security to information assurance. Traditional passive protection depends on static method such as firewall, vulnerability scanning, data encryption, access control and so on to reinforce the system. These security techniques all use predefined policy to response the attacks. Networks are multilevel and requirement for security are changing dynamically. Passive protection can't adapt to this new situation, so in-depth defense in military field is used to provide information assurance for complex networks. Proactive defense integrates the in-depth defense and extends the prediction and response to provide more agile and more effective control on system security.

Proactive Defense System (PDS) shows the following notabilities compared with traditional security system. (1)Openness, PDS is "open" to attacks outside and inside of the system, it even make attacks have some result. (2) Activeness, PDS predicts future intrusions or trends. Therefore, it can take actions in advance to eliminate or reduce the threat of attacks. PDS make counterattacks to some intrusions, this make it aggressive to some extent. (3)Dynamics, in the light of current security statistics, PDS predicts, assess and response intrusions dynamically. PDS extends intrusion protection to intrusion prediction, and enhances intrusion response system to provide active, dynamical and rounded information assurance service.

On the background of in-depth defense for network security, taking proactive defense model as research objective and security critical system as application environment, we explore proactive defense for its supporting theories.

2 Related Works

The development of vulnerability scanning and intrusion detecting system (IDS) provides technologic foundations for building dynamic security models. A PDR model is referred in the document of the US Department of Defense (DoD Directive S-3600.1, Information Operations). PDR puts emphasis on defending and recovering throughout the lifecycle. Main limitations of the succeeding dynamic security model P2DR involve (1) lacking early warning: It does not take prewarning stage of security period into consideration and is unable to predict intrusion. Therefore, it always falls one pace behind attacks with passive detection and reaction; (2) not dynamics in true meaning: The dynamics of P2DR grounds on intrusion detection. What's more, it adopts predefined responding policies, but not dynamic response according to the situation of intrusion threat; (3) Too coarsely granular security policy: P2DR defines policies for every security product, but not for the system's defending objective. This results in each security component works individually and policies have no association between each other, hence no coordinate safety control mechanics can be established.

Both of the initiation and dynamics are what the dynamic security model lacks. By now, there is not any security model that reflects the character of proactive defending. Besides traditional defending and detecting techniques, proactive defending related techniques also contain intrusion predicting and responding techniques.

3 Issues in Policy-Tree Model

Security model analyzes system with formal definition. It exactly defines security architecture and functions of target system with abundant semantics. From the global and unitary view of system security, we propose policy-tree model to reflect the property of proactive defense. We define security policy in the model, analyze the completeness, correctness and consistency on policy and provide corresponding solutions. Then policy-tree and the operations are defined formally.

3.1 Security Policy

Security policy is the set of subjected rules driven by security requirement. It's security rule by which system actions should abide. The target of security policy is to prevent attacks and minimize the loss after being attacked.

Definition 1. Security policy T is a triple, which reflects a certain defense target, denoted as $T = (C, O, F)$. Where C is the set of vulnerabilities, $C = \{cve_1, cve_2, \dots, cve_n\}$, cve_i is the vulnerability reference defined in Common Vul-

nerabilities and Exposures (CVE, Ver.20030402), O is set of the protected targets, such as host, router, storage device and so on the system level objects, which is weighed as different level to differentiate importance to the system. Cartesian product $C \times O$ is regarded as defense target set. $F = C \times O \rightarrow D \times E \times P \times R$, is a partial function from two-tuples to quadruple. F is a map describing the rules while protecting, detecting, predicting and responding intrusions should conform to. P, D, E, R describes sub-policy in four security phases respectively. P is set of sub-policy for protection, and that there exists a total function $f_p : C \rightarrow P$; D is set of sub-policy for intrusion detection, and that there exists a total function $f_d : C \rightarrow D$; E is set of sub-policy for intrusion early-warning, and that there exists a total function $f_e : C \rightarrow E$; R is set of sub-policy for intrusion response, and that there exists a total function $f_r : C \times O \times D \times E \rightarrow R$.

Basing on the security policy defined above, sub-policies, which reflect different demands in each phase of a security period, are systemized in a coupled and cooperated way. A policy-tree model, system defense oriented, is proposed to for security requirement formalizing, network deploying and developing applications with auto-defending ability in the security critical environment.

3.2 Policy-Tree's Semantic Model

Z language is a kin of model oriented and normalized description language. Its formalized semantics is grounded on set theory and the first order predication logic [1]. The Policy-tree's definition in Z language is illustrated in Fig. 1, where tip is an empty tree and fork is a one-to-one mapping binary relation, which combines two trees into a new one. The semantic model of policy-tree is a tree, in which the essence of policy-tree is given using cycle definition. For better response, the policy-tree runs policy clustering, therefore, its node is a policy of four sub modes or reflects the class property of the policy clustering's result.

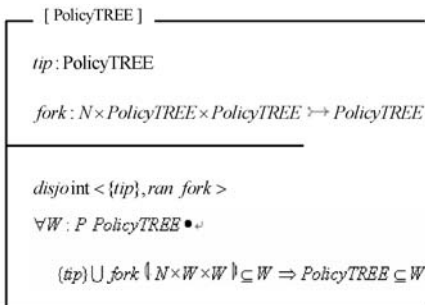


Fig. 1. Z Schema of Policy Tree

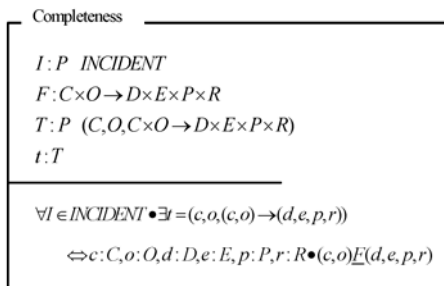


Fig. 2. Definition of Completeness of Security

3.3 Properties of Policy-Tree

The formalized definition of security policy has three basic requirements involving completeness, correctness and consistency [2]. In the initiative defending model, the following three problems are involved. First, whether the defined policy covers all the defending targets; second, is the policy practicable, namely, it does not contract with the experience; third, are there any policies conflict in the policy model, which may result in different decisions when encountering intrusion. Hence, it is necessary to define and explain the meaning and related properties of completeness, correctness and consistency.

3.3.1 Constitute the Completeness

In the definition of completeness (fig.2), INCIDENT denotes the universal set of intrusion event (n repeatable set). F is the rule set, standing for hex-tuple relation. T is the policy set and t is a single policy. The Completeness shows that any intrusion event can find out an according policy in the policy set. It is possible that some intrusion event policies are undefined in the complex network. Especially, it is true for unknown attacking events. Constructing system default policy can satisfy the requirement of Completeness. Namely, set a default responding policy, besides the established known defending targets-oriented policy, to handle the intrusion events outside the defending target set.

When a policy set is complete, the policy-tree can response to any intrusion events and it is also complete.

3.3.2 Validate the Correctness

The correctness of a policy is defined as follows (Fig. 3). One policy t corresponds to defending target (c, o) and rule $(c, o) \rightarrow (d, e, p, r)$. We say the policy t is correct when t is applied and the threat evaluation for the c type attack does not beyond the upper bound. Otherwise, t is incorrect. The in-depth analysis for intrusion and attack is the basis of making correct security policy. Meanwhile, the threat evaluation function and upper limit value also determine the correctness. For specific intrusion, proper responding policy should be chosen. It is supposed to response to intrusion quickly and to restrict the risk within an acceptable region. This indicates that it is not practical to eliminate all the risk in the network, but ease and control the risk.

3.3.3 Consistency Theory

Is it possible that conflict policies exist in the policy-tree? When *cve* vulnerabilities are common, c, p, e are the same while d, r may be different, because there are more one attacks aiming at one vulnerability. If r is different, then at least the intersection must be nonempty. r varies with the real situation, such as attack times, target range and the importance of object attacked. Hence choosing r is dynamic. There may be two policies, having the same defending target but different responding sub-policy. Here, the conflict means when *cve* vulnerabilities are the same and the intersection of protected objects is nonempty, responding sub-policy makes conflict responding measures, for example, packet blocked and passing allowed.

Correctness
$T: P \ (C, O, C \times O \rightarrow D \times E \times P \times R)$ $t: T$ $AssessRisk: C \times O \times D \times E \times P \times R \rightarrow N_i$ $limit: N_i$
$\forall t \in T \wedge t = (c, o, (c, o) \rightarrow (d, e, p, r)) \bullet$ $AssessRisk(c, o, d, e, p, r) \leq limit$

Fig. 3. Definition of Correctness of Security

Consistency
$T: P \ (C, O, C \times O \rightarrow D \times E \times P \times R)$ $\underline{CR}: R \times R \rightarrow \{0, 1\}$
$dom \ f_p = C \wedge$ $dom \ f_d = C \wedge$ $dom \ f_e = C \wedge$ $dom \ f_r = C \times O \times D \times E \wedge$ $\forall t_1, t_2 \in T \wedge t_1.c = t_2.c \wedge t_1.o \cap t_2.o \neq \emptyset \Rightarrow t_1.r \cap t_2.r \neq \emptyset \bullet$ $t_1.r \underline{CR} t_2.r$

Fig. 4. Definition of Consistency of Security Policy

Consistency is defined as fig.4. For any two policies t_1, t_2 , if $t_1.c = t_2.c$, $t_1.o \cap t_2.o \neq \emptyset$, then $t_1.r \cap t_2.r \neq \emptyset$ can be obtained from the definition of responding sub-policy (Same to $t_1.e = t_2.e, t_1.p = t_2.p$). If $t_1.r$ and $t_2.r$ satisfy CR relationship, we call policy set T is consistent, otherwise T is inconsistent.

CR is the consistency relation that defined in responding sub-policy. If $r_1 \underline{CR} r_2$ holds, (which shows that r_1, r_2 make responding measures without conflict), then the responding policies r_1 and r_2 is consistent. We can see from the definition of policy's consistency that there may be more than one policy for one defending target $< cve_i, o >$.

According to the definition of security policy's consistency, following two theories on consistency can be obtained.

Theory 1. If any two policies t_1, t_2 in a policy set T satisfy the following conditions that $t_1.c \neq t_2.c$ or $t_1.o \cap t_2.o = \emptyset$, then T is consistent.

Proof: $\forall t_1, t_2 \in T, t_1.c \neq t_2.c$ or $t_1.o \cap t_2.o = \emptyset \Rightarrow$ in T , all the defending targets are different. $\forall I \in P \text{ INCIDENT}$, if there exists according security policy to defending target of the intrusion event in the policy set, then this policy is unique. Therefore, no security policy conflict will occur and T is consistent.

According to theory 1, following corollary can be developed.

Corollary 1. Confliction occurs in the security policy only if cve vulnerabilities are the same and the protected objects' intersection is nonempty.

Theory 2. For any two policies t_1, t_2 in policy set T , if $t_1.c = t_2.c \wedge t_1.o \cap t_2.o \neq \emptyset$ holds, then T is consistent if and only if $t_1.r \underline{CR} t_2.r$ also holds.

Here, CR is the relation defined in the responding sub-policy set and it represents that the responding measures in two responding sub-policies do not conflict.

Proof: Sufficiency $\Rightarrow$

Confliction occurs in the security policy only when the *cve* vulnerabilities are the same in the defending targets and the protected objects' intersection is nonempty (Corollary of theory 1). Examine the parts in the policy set where conflictions most probably occur. The policy set T' is formed while policies with the same *cve* vulnerability and protected object are removed from original policy set T . $\forall t_1, t_2 \in T'$, if $t_1.c = t_2.c$ and $t_1.o \cap t_2.o \neq \emptyset$ holds, plus the condition $t_1.r \underline{CR} t_2.r$, the conclusion that T' is consistent, then T is also consistent can be drawn, according to the definition of policy's consistency.

Necessity $\Leftarrow$

According to definition 1, there exists total function $f_E : C \rightarrow E$; $f_P : C \rightarrow P$; $f_R : C \times O \times D \times E \rightarrow R$. $t_1.c = t_2.c \Rightarrow t_1.e = t_2.e, t_1.p = t_2.p, t_1.r \cap t_2.r \neq \emptyset$. Hence the two quintuples corresponding to the security policy are intersected. If $(t_1.r, t_2.r) \notin CR$, then the existence of same *cve* vulnerabilities, nonempty protected object intersection and disaccord security policy of responding sub-policy $\Rightarrow T$ is disaccord. This results in contradiction. Hence, if T is consistent, then the $t_1.r \underline{CR} t_2.r$ must holds.

Policy-tree contains thousands of security policies. Consistency of policy's definition is the premise of policy-tree model's implementation and application. It is definitely necessary to design auto-checking methods for consistency when facing huge policy set. Suppose that there are n security policies in policy set T .

Definition 2. The adjacency matrix of set $A = \{a_1, \dots, a_n\}$ in relation Q is denoted as $M_A = (m_{ij})_{n \times n}$, where $m_{ij} = \begin{cases} 1 & (a_i, a_j) \in Q \\ 0 & (a_i, a_j) \notin Q \end{cases}$.

Definition 3. Let the Boolean product of matrices be denoted as $A_{m \times n} \otimes B_{m \times n} = C_{m \times n}, c_{ij} = a_{ij} \otimes b_{ij}$; the Boolean add of matrices be denoted as $A_{m \times n} \oplus B_{m \times n} = D_{m \times n}, d_{ij} = a_{ij} \oplus b_{ij}$, where $\otimes$ is binary product and $\oplus$ is binary add.

Adjacency matrix and Boolean product of matrices are used to describe the consistency conditions of security policy. And Boolean add of matrices is utilized to compare the difference of two matrices and determine the conflicting policies in the policy set.

Auto checking method:

Denote the policy set $T : P(C, O, C \times O \rightarrow D \times E \times P \times R)$. Let $T = \{t_1, \dots, t_n\}$. C is vulnerability set. O is protected object set. R is responding sub-

policy set. $t_i.c, t_i.o, t_i.r$ stand for the vulnerability of policy t_i , protected object set of policy t_i and responding sub-policy set of t_i respectively.

a. Construct adjacent matrices M_C, M_O, M_R

$M_C : T.C$'s adjacent matrix with equal relation, $(\alpha_{ij})_{n \times n}, \alpha_{ij} = \begin{cases} 1 & t_i.c = t_j.c \\ 0 & t_i.c \neq t_j.c \end{cases}$

$M_O : T.O$'s adjacent matrix with intersected relation, $(\beta_{ij})_{n \times n}, \beta_{ij} = \begin{cases} 1 & t_i.o \cap t_j.o \neq \emptyset \\ 0 & t_i.o \cap t_j.o = \emptyset \end{cases}$

$M_R : T.R$'s adjacent matrix with CR relation, $(\gamma_{ij})_{n \times n}, \gamma_{ij} = \begin{cases} 1 & (t_i.r, t_j.r) \in CR \\ 0 & (t_i.r, t_j.r) \notin CR \end{cases}$

b. Let $M_1 = M_C \bar{\otimes} M_O, M_2 = M_1 \bar{\otimes} M_R \oplus M_1$. Matrix $M_2 = (\varphi_{ij})_{n \times n}$ stands for the result of consistency checking: ① if $M_2 = 0_{n \times n}$ (all-zero matrix), then the policy set T is conflict less; ② if M_2 is non-zero matrix, then conflicts exist in policy set and the subscripts whose items equal to 1 in M_2 stand for the numbers of conflicted policies. For example, $\varphi_{ij}=1$ indicates that t_i, t_j is disaccord in policy set T.

Proof for the conclusion goes like this,

Let $M_1 = (\theta_{ij})_{n \times n}$. According to the definition of matrix Boolean product, assign θ_{ij} the value 0 or 1. When $\theta_{ij} = 0, \Rightarrow cve$ vulnerabilities of policies t_i and t_j are different, or their protected objects' intersection is nonempty, namely, $t_i.c \neq t_j.c \vee t_i.o \cap t_j.o = \emptyset$. Basing on Theory 1, we can infer that policies t_i and t_j are consistent. No matter what value of γ_{ij} is, $\theta_{ij} \bar{\otimes} \gamma_{ij} \oplus \theta_{ij} = 0$ and $\varphi_{ij} = 0$ hold.

If $\theta_{ij} = 1, \Rightarrow cve$ vulnerabilities of policies t_i and t_j are the same and their protected objects' intersection is nonempty, namely, $t_i.c = t_j.c \wedge t_i.o \cap t_j.o \neq \emptyset$. According to Theory 2, if t_i and t_j are consistent, then $(t_i.r, t_j.r) \in CR$ should holds, namely, $\gamma_{ij}=1$, and now, $\theta_{ij} \bar{\otimes} \gamma_{ij} \oplus \theta_{ij} = 0, \varphi_{ij} = 0$ hold.

If $\theta_{ij} = 1, \Rightarrow t_i.c = t_j.c \wedge t_i.o \cap t_j.o \neq \emptyset$. Moreover, if $\gamma_{ij}=0$ also hold, it can be inferred that cve vulnerabilities of policies t_i and t_j are the same and their protected objects' intersection is nonempty but there are conflicts in responding sub-policy. Applying Theory 2, we can know that policies t_i and t_j are conflicted. Now, $\theta_{ij} \bar{\otimes} \gamma_{ij} \oplus \theta_{ij} = 1$ and $\varphi_{ij} = 1$ hold.

In all, when M_2 is zero-matrix, the policy set T is consistent. When M_2 is non-zero-matrix, its subscripts whose items equal to 1 stand for the numbers of conflicted policies.

The computational complexity analysis for consistency checking algorithm goes as follows. For a policy set with n security policies, constructing M_C, M_O, M_R , requires $n(n-1)/2$ times comparing operations, involving determining whether two numbers are equal, whether two sets are intersected, or whether two responding measures are consistent. M_C, M_O, M_R are symmetric matrices, namely, $\alpha_{ij} = \alpha_{ji}, \beta_{ij} = \beta_{ji}, \gamma_{ij} = \gamma_{ji}$. In the b steps of the algorithm, $n(n-1)$ times bit-product and $n(n-1)/2$ times bit-add are required to determine whether there are conflixtions and the positions of conflicted policies.

ConsistencyChecker is illustrated in fig.5. Adjacent matrix function adj of set A with relation R is defined at the beginning. The inputting of ConsistencyChecker schema is sequence T, and the output is matrix M, standing for the consistence. Matrices in the schema are represented as sequences.

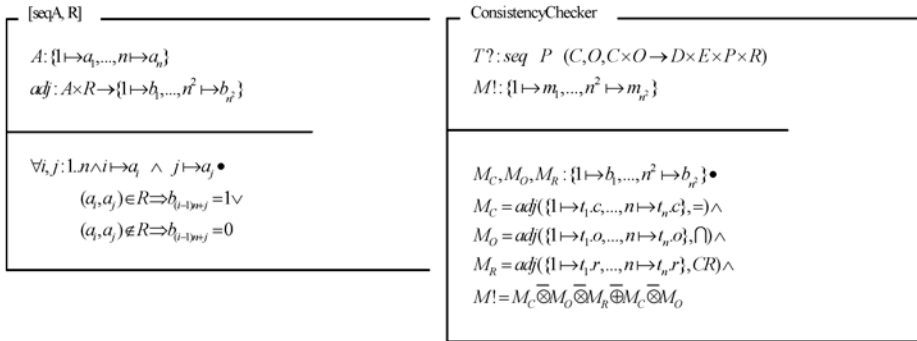


Fig. 5. Consistency Checker for Security Policy

Utilizing ConsistencyChecker, we can easily find out the maximum non-conflict subset in a given policy set T. Specifically, apply ConsistencyChecker to T and get M_2 , cross out the rows and columns whose items include 1 in M_2 , keep the left items' original numbers. Then an all-zero matrix M_2' is obtained. No matter which column is picked out from M_2' , the sequence of its subscripts' column number is the numbered sequence of policies contained in the maximum non-conflict subset in T.

3.4 Basic Operations of Policy-Tree

Basic operations of policy-tree include search, add, delete, replace, and combine. Search does not modify policy-tree's status space, while add, delete, replace and

combine will result in the change of policy-tree. Modifying the policy-tree may cause the inconsistency. Therefore, checking for policies' consistency is necessary. Here, we provide the formal definition of Z schema for policy-tree's basic operations. Declare the status schema of policy-tree *PolicyTree* as follows. *known* is the two-tuples set (C,O) of existing policies in the policy-tree. It stands for the policy's defending targets. T denotes the policy set of policy-tree.

3.4.1 Create

The policy-tree has treelike structure with depth 4. Its first layer is root node, the access entrance of policy-tree. The second layer contains category nodes, which can be classified into M broad headings by attacking properties [3]. The third one contains policy nodes, which specify the defending targets of security policies and whose father nodes denote the attacking types. The fourth one contains leaves, which stand for sub-policies against the defending targets and whose father nodes denote defending targets.

Tree creating establishes the three layer nodes and places all the inputted policies at the according nodes in the policy-tree in response to the categories they belong to, maintaining the consistency of the whole policy-tree. Layer-by-layer evolving method is adopted to create policy-tree. First, all the nodes on the *i* th layer are established, then turn to the *i*+1 th layer. In the tree creating procedures described below, please refer to fig.5 for ConsistencyChecker, and the operation of add policy *AddPolicy* to policy-tree is specified below.

```

Create a tree: CreatePTree
Input: root node root, security policy set S, attacking type set C
Output: A policy-tree with depth 4 and consistent policies

1. Initiate root node;
   1.1  $K \leftarrow \emptyset$ ; //node set contained in the tree
   1.2  $U \leftarrow S$ ; //node set to be added to policy-tree
2. for  $i = 1$  to  $m$  //create attacking type nodes on the second layer
   2.1 Pick one classname from C and set to  $c_i$ ;
   2.2  $Class[i] \leftarrow c_i$ ;
3. while  $U \neq \emptyset$  do //create policy nodes and their leaves
   3.1 Pick one policy and set to  $t_i$ ;
   3.2  $flag \leftarrow ConsistencyChecker(\{t\} \cup \bigcup_{\substack{t_k \in K \\ t_k \in class \neq class}} t_k)$ ;
   3.3 if  $flag = 0$  then AddPolicy( $t_i$ , root); //Add policy  $t$  to policy-tree
   3.4  $U \leftarrow U - \{t\}$ ,  $K \leftarrow K \cup \{t\}$  //Modify  $U$  and  $K$ 

```

3.4.2 Search

The searching operation locates one policy in a policy-tree according to policy defending target (*cve* vulnerabilities and protected objects). Function input: vulnerability *cve_i*, protected object set *o* (systemic level); Output: policies (protected object set, sub-policy sets of detecting, predicting, defending and responding), result. If the policy node that satisfies the condition is empty, then return "unknown".

In the vulnerability library CVE (Ver.20030402), 2572 kinds of hole are defined. Considering that there may be several policies corresponds to one *cve* vulnerability, larger numbers of policies should be contained in the policy-tree. Moreover, a kind of storing structure need well designing to support the fast searching for security policies, because protecting, detecting, predicting and responding to intrusion events all depend on security policies. This structure will affect the performance of policy-tree wholly.

According to the policy-tree's structure defined in section 3.3, security policies are divided into m types basing on the *cve* vulnerabilities involved. Suppose that the i th type contains N_i ($i = 1, 2, \dots, m$), if we search by sequence, then the first step is to find out the catalog policies belongs to and the second step is to search policies according to *cve* vulnerabilities by sequence. The average searching step size is

$$\frac{1}{m} \sum_{i=1}^m \left(\frac{m}{2} + \frac{N_i}{2} \right) = \frac{m}{2} + \frac{1}{m} \sum_{i=1}^m N_i / 2$$

Considering that the magnitude comparison is applicable to both of them, the policy type and *cve* vulnerability numbers can be indexed. In this way, binary search can be applied to both catalog search and number search for *cve* vulnerabilities within the catalog. Hence, the average searching step size is

$$\frac{1}{m} \sum_{i=1}^m (\log_2^m + \log_2^{N_i}) = \log_2^m + \frac{1}{m} \sum_{i=1}^m \log_2^{N_i}$$

Efficiency of the algorithm is greatly improved. The storage of policy-tree is illustrated in fig. 6 in the form of two-dimensional chain.

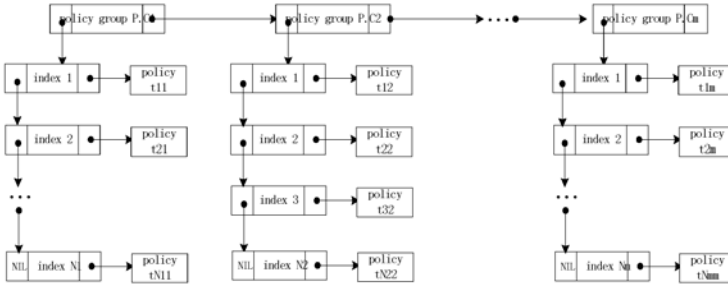


Fig. 6. Two-dimensional Link Storage Structure for Policy Tree

3.4.3 Add

To add a new policy t to policy-tree (fig. 7), policy consistency checking must be done first. After being classified by policy-tree, *cve* vulnerabilities according to different kinds of policies are definitely different. Therefore, the consistency checking can be carried out within the same kinds of policies. This reduces the checking space significantly. Once there is no confliction in the policies, place policy t at the leaf where it belongs to that catalog. If such policy has existed in the policy set, return “known”. Otherwise, return “inconsistency”.

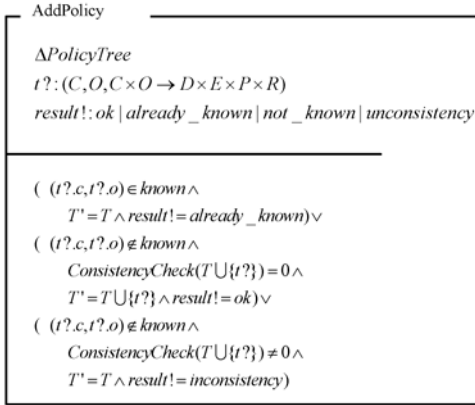


Fig. 7. AddPolicy Schema

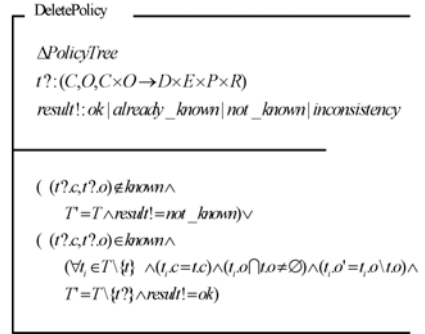


Fig. 8. DeletePolicy Schema

3.4.4 Delete

Deleting one policy t means that its according sub-policies of protecting, detecting, predicting and responding are no longer valid. If simply remove t from policy set T , then $T - \{t\}$ may still contains policies whose protecting objects are consistent with policy t . This will result in incomplete deleting. Therefore, besides deleting t , policies in $T - \{t\}$ whose protecting objects have intersection with policy t 's must be modified. $\forall t_i \in T - \{t\}$, policy t_i that satisfies $t_i.c = t.c \wedge t_i.o \cap t.o \neq \emptyset$ should be modified as $t_i.o' = (t_i.o \setminus t.o)$. Delete operation shown as fig. 8.

3.4.5 Replace

Replacing operation means to replace existing policy t_1 with a new policy t_2 . Here, policies t_1 and t_2 have the same defending target ($t_1.c = t_2.c \wedge t_1.o = t_2.o$), however, their sub-policies of protecting, detecting, predicting and responding are different ($t_1.p \neq t_2.p \vee t_1.d \neq t_2.d \vee t_1.e \neq t_2.e \vee t_1.r \neq t_2.r$). Replacing operation equals two steps. First is to delete t_1 from policy set T and second is to add t_2 . Hence, we can obtain replacing schema via combining deleting schema and adding schema. That is, $ReplacePolicy \hat{=} DeletePolicy \wedge AddPolicy$.

One thing must be point out that replacing one policy in a policy-tree is not equivalent to replacing policy nodes in the tree directly, because in the first step (deleting old policies), other policies may be modified and this is caused by assuring the consistency.

3.4.6 Combine

Policy combining (fig.10) means merging two policies t_1, t_2 in T , if they satisfy $t_1.c = t_2.c$. It maintains *cve* vulnerability unchanged. The protected objects are the

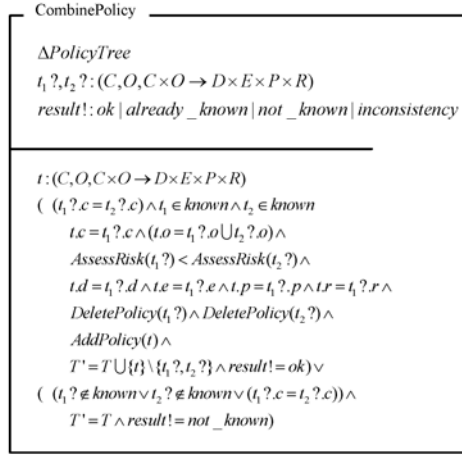


Fig. 10. CombinePolicy Schema

union of the two and sub-policy with smaller threat assessment value is chosen. In the policy merging schema, if t_1 and t_2 exist in policy set and $t_1.c = t_2.c$ holds, then the two are consistent as well as with other policies in T . New policy $t.r$ obtained in this way is selected from $t_1.r, t_2.r$. According to Consistency Theory 2, it is easy to conclude that t is also consistent with $T \setminus \{t_1, t_2\}$. Therefore, policy combination can be exempted from consistency checking, if two policies satisfy that t_1 and t_2 exist in policy-tree, and $t_1.c = t_2.c$ holds.

4 Conclusion

Besides protection and detection of traditional defense models, the proactive defense model adds early warning and dynamic response. This model based on policy-tree improves system's activity, adaptability and survivability via proactive defense, dynamic detection, active prediction and response. Dynamic security policy goes throughout the security period. The function of security policy is quite different from dynamic security model, such as P2DR. The former aims at defended targets and possesses more activity and dynamic characters, while the latter aims at specific security products and has passive defending ability and partial dynamics. Complete technical system calls for further works.

Acknowledgement

The authors thank for Mr. Shiming Yang's helpful discussion on matrix theory and his typing of this paper.

References

1. J. M. Spivey. The Z Notation, A Reference Manual. International Series in Computer Science, Second Edition. Prentice-Hall International, 1992.
<http://spivey.orient.ox.ac.uk/~mike/zrm/zrm.pdf>
2. Li Shoupeng, Sun Hongbo. Security Policies Study for Information Systems. [J]Acta Electronica Sinica. 2003, 31(7): 977-980.
3. The Snort Project. Snort Users Manual.
http://www.snort.org/docs/snort_manual.pdf. Dec,2003. p22-24
4. T. Tidwell, R. Larson, K. Fitch et al. Modeling Internet Attacks. Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, 2001: 54-59.
http://www.itoc.usma.edu/marin/Wshop/Papers2000/TP1_4.pdf

Researches on Scalable Architecture for Security Information Distribution Service*

Haitao Chen, Chuanfu Xu, Zunguo Huang, Zhenghu Gong, and Hua-Ping Hu

School of Computer Science, National University of Defence Technology
Changsha, Hunan, China
nchrist@163.com

Abstract. Distribution of security information is one of the important infrastructures for information security of Internet. There are some problems in the existing security information distribution (SID) Services on Internet such as single point of failure, denial of service under flash crowd and the bad time efficiency of information distribution. In this paper we propose two scalable architectures for SID service based on peer-to-peer technology and apply them to the SID service of the China Education and Research Network (CERNET).

1 Introduction

The Internet has become the infrastructure of modern information society and has deeper and broader impact on social development. At the same time the security threats that Internet faces are becoming more and more serious. Especially the worm-like attacks that can spread quickly and broadly challenge the Internet security defense. High speed SID service with strong survivability is needed to solve this challenge. This service is responsible for maintaining the security robustness of network application by sending security advice and security patches to the possible attacked object before the arrival of viruses. There are some problems in the existing SID services on Internet such as single point of failure, denial of service under flash crowd, the bad time efficiency of information distribution.

In this paper we propose the construction of SID service based on P2P technology. The design of the service includes following ideas: distribution of network application to multiple heterogeneous servers to improve the availability of the application; organization of clients into peer-to-peer cooperative network to deal with flash crowd by load-aware content replicating algorithm; fast push of security information independent of the support of multicasting of network layer on Internet by the collecting property of logic P2P network.

* This project is supported by the National High-Tech Research and Development Plan of China under Grant No.2003AA142080, the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314802.

2 SID Based on HTTP Access Mode

Assumption of Application Scene

- A group of servers scatter about the Internet with cooperation intension.
- Servers provide SID service.
- Clients can not be modified.
- The amount of clients is huge.
- The access of clients has flash crowd.

Architecture Design

Servers form a DHT network for the purpose of service cooperation. In user's view, it's a heterogeneous service cluster on wan network without the help of front end dispatcher. In the DHT network, servers can put their heavy load contents into other servers based on some rules and redirect their clients' requests to cooperating servers under heavy load. We can see the architectures in figure 1.

Components

- Classification of URL: We classify URL into three types based on the method needed to resolve them. Heavy load URLs are URLs which constitute the main accessing load of servers like pictures, big files and so on. Light load URL are local URLs which may not lead servers to be overloaded. Load-collaborating URLs are temp URLs used to provide access service for the content of other servers for the sharing of load.
- URL resolver: It resolves clients' requests according to the types of URLs. For heavy load URLs, it redirects the URL according to load balance protocol. For light load URLs, it executes normal process. For load-collaborating URLs, it reads corresponding content according to URL transform protocol.
- Virtual publishing pool: Each server maintains a virtual publishing pool and puts heavy load content into this pool. The content in pool will be distributed to other servers' load-collaborating pools according to peer-to-peer protocol.
- Load balance scheduler: It executes schedule policy based on current load of servers.
- Load-collaborating pool: Each server maintains a load-collaborating pool which holds content coming from other servers.

Access Flow

- 1) Servers publish contents in virtual publishing pool in the P2P network composed of servers.
- 2) Client accesses one server by HTTP protocol.
- 3) The accessed server checks its load. If the load is low, it will service the client directly, or it will redirect this client to other server with low load according to the load balance algorithms. The redirection URL is produced in accordance with URL transform protocol.
- 4) Client gets what it needs by the new URL.

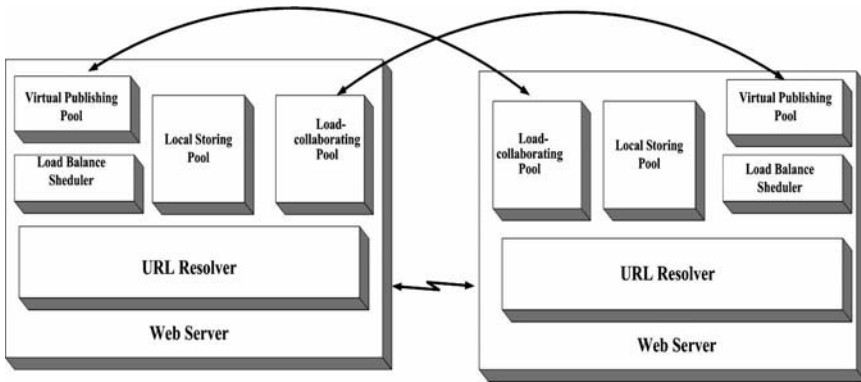


Fig. 1. Architecture of servers

Load-Aware Content Replicating Policy

The content replicating policy has great influence on system performance. We design a load-aware replicating policy. If the number of servers is N , the policy classifies the load of server into $\lceil \log_2 N \rceil$ grades. When the load grade of the server is K , the number of content replication is 2^K . The load balance scheduler controls the number of replication based on the load of server.

3 SID Based on P2P Access Mode

3.1 Assumption of Application Scene

- A group of servers scatter about the Internet with cooperation intension.
- Servers provide security information downloading and broadcast service.
- Clients can be modified.
- Huge amount of clients exist in multiple LANs with local server.
- The access of clients has flash crowd.

3.2 Architecture Design

We design a two-layer P2P network to meet the above requirements. There are researches about the hierarchical P2P network in [1]. As in figure 2, the architecture includes two kinds of P2P network.

Server P2P network: The servers compose a DHT P2P network. Servers implement load balance by P2P collaboration.

Local P2P network: The clients in one LAN and local server compose local DHT P2P network. Local server attends both the up-layer server P2P network and down-layer local P2P network. Local server is the boot-strap node and the manager of local P2P network. Local P2P network ensures the high download speed of clients and distributes the security information to clients.

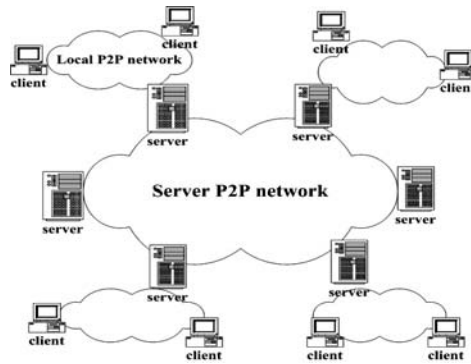


Fig. 2. The sketch map for architecture of hierarchical SID

3.3 Security Information Publishing and Intelligent Downloading

Load-Sharing Flow

- 1) Servers make index file of original file and publish them on the Web. The index file includes information about how to download the related file on P2P network.
- 2) Client accesses server by HTTP mode. It clicks the downloading hyperlink of P2P mode and downloads the index file.
- 3) The plug-in of the client's IExplorer reacts to this event of P2P downloading request and submits the event to P2P software of client.
- 4) P2P software searches the file in local P2P network. If it finds the file, it will download the file in P2P manner and switches to step 8.
- 5) P2P software makes requests to the local server for the file.
- 6) Local server searches the file in server P2P network and downloads it.
- 7) Client P2P software downloads the file from local server.
- 8) Client P2P software publishes the file in local P2P network.

Intelligent Downloading Scheduling Policy

The downloading schedulers work in both server P2P network and local P2P network. Because the scheduling algorithms are similar, we take the scheduling algorithm of local P2P network for an example.

Each file published in P2P network has a master node that is chosen by DHT algorithm and is in charge of scheduling of downloading the file. All nodes that hold the file will register correlative information into the master node. As to the whole local P2P network, the downloading of different files is distributed scheduling process. But as to any single file, the downloading scheduling is centralized.

The master nodes schedule the file downloading based on these rules:

- Low load nodes and neighboring nodes have higher priority.
- Client can download from multiple nodes at the same time.
- The big file with high load will be split into multiple slices and be downloaded from multiple places.

3.4 Security Information Broadcast

In order to control the spreading of worm, the security updating information like operating system security patch must spread quicker than the worm. P2P network has a good support for broadcast of application layer and can be used to implement fast security information broadcast.

Security Information Broadcast Flow

- 1) Broadcasting the indexing file of security information: First the index file is broadcasted in the server P2P network. Then each local server broadcast the index file in local P2P network.
- 2) Filtering security information: Here we have two implementing choice. One is distributed decision-making. The client will make decision on whether to download the security information or not based on the description of security information and local security status. The other is centralized decision-making. Local server collects the security status of local clients and makes decision.
- 3) Downloading: Clients start the downloading process. The security information spread first in up-layer network, then in the down-layer networks.

4 Advantages Analyzing

Compared with the existing SID services, our design based on P2P technology has some advantages.

4.1 In-Depth Load Balance

The two access modes both inherit good load balance of DHT, and implement higher load balance according to properties of application layer.

Our first design implements the load balance of servers that takes good use of the servers' capacity. The second design implements the load balance of servers and also the load balance between clients. They solve the problem of the demand and supply of network bandwidth.

4.2 Adaptability

- The number of participating nodes can change dynamically. Nodes can join or leave casually, and the number of collaborating nodes can change dynamically in the two access modes. This is supported by dynamical adaptation of P2P network.
- Collaborating information is distributed dynamically. Information published in the P2P overlay network can change and adjust dynamically to adapt to the variability of security information.
- Collaborating relation is adaptive. Collaborating relation of participating nodes can adjust according to predefined security policy and load of nodes without of intervene of users.

- Information replication is load-aware. Servers in the first access mode can control replication according to its load and it is a centralized policy. Information in the second access mode will be distributed automatically under heavy load and it is a decentralized policy. Load-aware information replication is realized in both two access mode, namely information replication can adjust according to magnitude of access. Load-aware information replication not only meets the demand of obtaining information quickly and instantly, but also decreases usage of resource.

4.3 Survivability Analyzing Under Attack

- Analyzing of time validity for broadcasting security information: The time for broadcasting security information is logarithmic to number of nodes, which means good scalability [2]. Broadcast of security information will be ended in valid time.
- Survivability under flash crowd: Both design have good load balance algorithm and can work well under flash crowd. We will make quantitative analysis in further paper.
- Excursion of access under node failure: Single point of failure has no too much affection in the first access mode; and servers will form a server cluster without single point of failure when all servers provide same service. Single point of failure has no impact on downloading and broadcasting in the second access mode, and service can shift to other service node automatically.

4.4 Explicit Incentive of Nodes

The incentive of node participating in the P2P community is basis for continual and stable running of the P2P overlay network. Server nodes participate for collaboration to lessen load and client nodes can get security information instantly to lower its security risk through participating.

5 SID Service of CERNET

Requirements Analyzing

The research of this paper is applied and verified in the security information service of CERNET. CERNET is composed of multiple connected LAN and backbone network. Its security service is responsible for distributing security notification, security patch, and security knowledge to the users of CERNET. The existing architecture is mainly a master server and local servers without cooperation. There are some problems like single point of failure, denial of service under flash crowd, the bad timely efficiency of information distribution for the existing SID services on Internet.

Architecture Design

Based on the requirements of CERNET, we design a new architecture by integrating the above two architectures which you can see in figure 3. According to the architecture one, the servers compose a P2P network and provide browsing and downloading service to unattached and decentralized clients as a web cluster on WAN without the

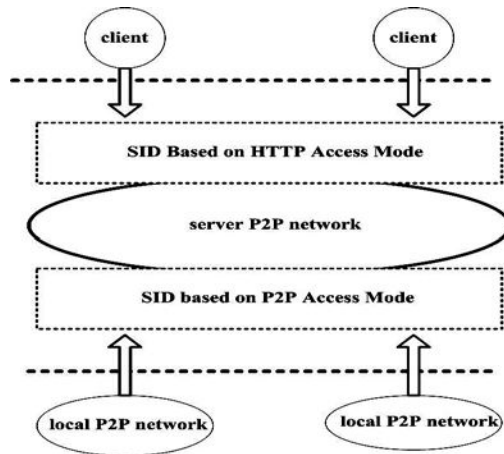


Fig. 3. Architecture of SID for CCERT

help of front end scheduler. According to the architecture two, the servers and clients also compose a two-layer P2P network and provide downloading and broadcasting service to clients in the LAN.

Now the application of this integrating architecture is under construction. It will be an important supporting infrastructure for the security service of CERNET. We will refine the system on real running data.

6 Related Work

There are few researches on the large scale SID service. The existing systems like the updating service of Microsoft are mostly a cluster. There is no research work that is very similar to our work, but some related work about content publishing and locating exist.

Globule [3] proposes a cooperating method for web sites based on P2P manner. The cooperating site opens some resources for other sites. And in return it can store its content in other sites. Under heavy load, the original site will redirect the request to cooperating sites by DNS redirection or HTTP redirection. The cooperating and replicating policy are static and rely on manual configuration. Because the replication must be the whole site, the cooperating servers must be homogeneous.

Literature [4] proposes to use caches made of web clients to deal with flash crowd. The server registers the clients that are willing to be as caches. Being overloaded, the server will service clients' requests by returning a list of usable redirection caches. Then the client will choose the nearest cache client and redirect its request to the cache client. This method is lightweight and is only suitable to solve small-scale sites. Also the clients have no incentive to be a web cache.

7 Conclusions

In order to defense the security threat of Internet and solve the problems of the existing SID systems, we apply the P2P technology to the SID service and propose two novel architectures for SID service based on structured P2P network in this paper. Now we are analyzing and verifying these architectures more seriously by applying these architectures to the SID service of CERNET. We will refine the design based on experimental data.

References

1. Haitao Chen, Chuanfu Xu, Zunguo Huang, Huaping Hu and Zhenghu Gong: A Scalable Peer-to-Peer Lookup Model, Second International Workshop on Grid and Cooperative Computing(GCC2003), Shanghai, China: LNCS 3032, 2004, pp379-387.
2. Sameh El-Ansary, Luc Onana Alima2, Per Brand, Seif Haridi: Efficient Broadcast in Structured P2P Networks. IPTPS'03, Berkeley, CA, February, 2003.
3. Guillaume Pierre, Ihor Kuz, Maarten van Steen, Andrew S. Tanenbaum: Differentiated strategies for replicating Web documents[J], Computer Communications 24(2001) No.2, pp232-240.
4. Venkata N., Padmanabhan, Kunwadee Sripanidkulchai: The Case for Cooperative Networking[C], International Workshop on Peer-to-Peer systems (IPTPS 2002), Cambridge, MA, USA:LNCS 2429 ,2002, pp 178-190.

A Sequential Pattern Mining Algorithm for Misuse Intrusion Detection*

Shi-Jie Song, Zunguo Huang, Hua-Ping Hu, and Shi-Yao Jin

School of Computer Science, National University of Defense Technology
ChangSha, Hunan, China
songshijie87@vip.sina.com, zunguo@263.net

Abstract. This paper presents a sequential pattern mining algorithm for misuse intrusion detection, which can be used to detect application layer attack. The algorithm can distinguish the order of attack behavior, and overcome the limitation of Wenke Lee's method, which performs statistical analysis against intrusion behavior at the network layer with frequent episode algorithm. The algorithm belongs to behavior analysis technique based on protocol analysis. The preprocessed data of the algorithm are application layer connection records extracted from DARPA's tcpdump data by protocol analysis tools. We use vertical item-transaction data structure in the algorithm. Compared with AprioriAll algorithm, the complexity of this algorithm is decreased greatly. Using this algorithm, we dig out an "intrusion-only" itemset sequential pattern, which is different from normal user command sequential pattern. Experiments indicate that our algorithm describes attacks more accurately, and it can detect those attacks whose features appear only once. Our presentation offers a new approach for the research of misuse intrusion detection.

1 Introduction

As an important topic of information security assurance, intrusion detection has become research focus of network security. Resolving false-positive and false-negative of intrusion detection depends on the improvement of analysis technology. Currently the main intrusion detection analyses methods include pattern match, statistical analysis, protocol analysis¹ and behavior analysis². In addition, different from above traditional detection method, Wenke Lee research group³ of Columbia University of America has applied data mining to intrusion detection.

Pattern match detects attacks by matching attack signature exactly. The defects are over calculating quantity, and cannot detect transformable signature attack. Statistical analysis detects attacks by counting the correlation transactions of network instead of analyzing individual connection record, but does not considers the order of occurrence. Protocol analysis reassembles network data stream and parses application layer protocol, and then detects attack with pattern match and statistical analysis. This method can improve the accuracy and efficiency of detection. Behavior analysis can not only detect individual connection requirement and response, but also consider a session as whole. Some attacks cannot be detected in one connection requirement and

* This work was supported by 863 items No.2003AA142080 and No.2003AA142010.

response, because the attack behavior pervades in many connections. Thus, behavior analysis is becomes the trend of intrusion detection technology research. But, the algorithm and rule are complex and immature.

Wenke Lee applies data mining technique to intrusion detection, mines frequent episode rules from network layer training data, structures statistic features and builds classification detection model. If the training data include the new attack, the method can find out the rule of the new attack, thus we need not to download rule base from Internet. And the method will not be influenced by transformable signature attacks.

This paper first analyzes the main idea and deficiencies of Wenke Lee's method. Since the deficiency of network layer data, the application layer connection records that extracted from DARPA's tcpdump data by protocol analysis tools are used as the preprocessed data. We extend Wenke Lee's frequent episode algorithm, and present a misuse intrusion detection algorithm based on sequential pattern mining. Compared with AprioriAll algorithm, the complexity of this algorithm reduces tremendously. At last, we give the sequential pattern comparison and match algorithms, dig out a new "intrusion-only" sequential pattern, and implement a prototype of misuse intrusion detection. Currently, the research of IDS based on data mining is improved from other aspect of Wenke Lee's idea in domestic research.⁴ And to our knowledge, we haven't discovered the research of misuse IDS for application layer connection record based on sequential pattern mining overseas.^{5,6}

2 Wenke Lee's Main Research Idea and Deficiencies

Wenke Lee implements a network layer misuse IDS. His main work is to preprocess DARPA's tcpdump data with frequent episode algorithm³, and structure statistic features. And then he builds misuse intrusion detection model with RIPPER classifier. Whereas, there are several defects in Wenke Lee's method:

- Wenke Lee's main work is to mine frequent episode on network layer. The attacks detected by the method are almost probe and DOS attacks. These types of attacks have evident features that can be detected by many commercial IDS. But detecting R2L and U2R application layer attacks is the main focus of intrusion detection research at present.
- In order to detect application layer attacks, Wenke Lee selects statistic features with domain knowledge to detect attacks based on content. And there is no accurate criterion for the selection of the statistic features.
- And the most important, the character of statistical analysis can't reflect the relationship of context in the time order. But many intrusion behaviors depend on the time order. In that case, statistical analysis technique exists within severe limitation. E.g., the attacks based on content have not frequent connection records, and the features appear only once in an attack. We need to extract statistic feature from data with domain knowledge. So, using statistic method would lose order information. Because attack features are involved in many connection records, we need to find rules between connection records, and dig out itemset sequences of reflecting attack features, that is data mining technique based on sequential pattern. It is a severe deficiency because Wenke Lee's frequent episode algorithm cannot mine sequential pattern.

3 Sequential Pattern Mining

The user's normal behavior is very complex, it is difficult to describe it with several commands, while describing an attack is relatively easy. So, we build a misuse intrusion detection model based on sequential pattern mining.

Definition 1: sequential pattern

- Itemset: A non empty set is structured by all items.
- Sequence: Sequence is also called as itemset sequence. It is a row of ordered itemset. The itemset i can be shown as $(i_1, i_2, \dots, i_n)$, i_j means a item. A sequence s can be shown as $\langle s_1, s_2, \dots, s_n \rangle$, s_j means a itemset.
- Instance: All the records of an attack. Multi-instances mean to repeat the attack several times.
- Support of a sequence: The ratio between the number of instances that contain the sequence and the number of all the instances.
- K-large-sequence SL_k : The sequence that its support is bigger than given minimal support and sequence length is k . i.e., frequent k sequence.
- One-large-sequence- k -itemset L_k : A one-large-sequence that its itemset size is k , which is also called k -large-itemset in association rule.

Now we build a misuse IDS model based on sequential pattern mining algorithm. First, we get preprocessed data of application layer connection record sequence with protocol analysis tools. Then we emphasize to mine frequent itemset sequence from different instances of an attack. Comparing with frequent episode algorithm, sequential pattern mining algorithm can find out correlation of different connection records. And then we obtain "intrusion-only" pattern sequences by comparing normal sequential patterns. At last, we build an intrusion detection model with the "intrusion-only" sequential pattern tree, which is used to match the ready detection data.

3.1 Protocol Analysis Preprocessing

In the Intrusion Detection Evaluation project sponsored by DARPA, Lincoln laboratory of MIT offers tcpdump standard data⁷ for testing IDS. We preprocess tcpdump data with protocol analysis tool Net Monitor⁸. Then, we extract application layer connection record attributes out as items of sequence, and these items include command attribute and other attributes, which are shown as table 1.

Table 1. Application layer connection record attributes

Time	Source IP	Source port	Target IP	Service	Command	Requirement parameter	Response parameter	Sensitive information

In the table, sensitive information means the sensitive character string in the transmission data, which include accessing system sensitive directory and control files, e.g., "/etc/passwd", "/var/log", ".rhosts", finding compromised states on the destination host, e.g., file/path "not found" errors, and appearing a large amount of "NOP" instructions, etc.

3.2 Sequential Pattern Mining Algorithm Description

The aim of sequential pattern mining algorithm is to find out frequent itemset sequences from multi-instances of an attack. Compared with traditional Apriori algorithm, sequential pattern mining data are two more dimensions, one dimension is attribute value vector, and the other is time vector, i.e. attack instances.

The algorithm includes two steps, the *first step* is to search for one-large-sequence SL_1 :

1. Each item in original database is a candidate one-large-sequence-one-itemset C_1 . When we take out a C_1 , we vertically scan each item in original database. If an instance contains the C_1 , then the support of C_1 adds 1. And if the support is greater than the given minimal support, the C_1 is one-large-sequence-one-itemset L_1 . We find out all L_1 s in turn, and save all binary records of L_1 into temporary database.
2. We combine two one-large-sequence-(k-1)-itemset L_{k-1} s in temporary database to form candidate one-large-sequence-k-itemset C_k . We find out two L_{k-1} s with the same former k-2 bits, and combine the same former k-2 bits and two no. k-1 bits to form a C_k . In the C_k , the order of two no. k-1 bits are arranged as the original order.
3. When we take out a C_k , we vertically scan each L_{k-1} in the temporary database. And then, we horizontally search for the no. k bit. If it exists in an instance, the support of C_k adds 1. And if the support is greater than the given minimal support, the C_k is L_k . We find out all L_k in turn. And then list all one-large-sequence $SL_1, L_1, L_2, \dots, L_k$. Step 1 is shown in figure1.

The *second step* is to search for k-large-sequence SL_k .

1. We combine two k-1 large sequence SL_{k-1} s in temporary database to form candidate k-large-sequence SC_k . We find out two SL_{k-1} s with the same former k-2 bits, and combine the same former k-2 bits and two no. k-1 bits to form four SC_k . In these SC_k , the two no. k-1 bits can be arranged as four orders.
2. When we take out a SC_k , we vertically scan each SL_{k-1} in the temporary database. And then, we vertically search for the number k bit. If it exists in the instance, the support of SC_k adds 1. And if the support is bigger than given minimal support, the SC_k is SL_k . We find out all SL_k in turn. And then list all large sequence $SL_1, SL_2, \dots, SL_k$. Step 2 is shown in figure 2.

In order to describe the algorithm clearly, we will take the example of an attack includes 10 instances, each instance includes 100 connection records, each record includes 8 attributes, and each attribute includes 8 values. We save each attribute value of each record of all the attack instances into a raw data table in binary for data mining, which is shown in figure 3. We save all the one-large-sequence into D_T table in binary as mining data of next step, which is shown in figure 4. We get large amount of sequential patterns with the algorithm, which is shown in figure 5.

step is scanning database when counting support. Thus, our algorithm reduces complexity of sequential pattern mining algorithm greatly.

When searching for L_1 with Apriori algorithm, we need to scan the database once, and count all items (i.e., C_1) that appear in all connection records. When searching for L_2 , we need to scan the database once again, and count all C_2 that appear in all connection records. After finding out L_k , we scan the database k times totally.

When we search for SL_1 , the AprioriAll algorithm is the same as Apriori algorithm. When searching for SL_k , we need to scan all the k connection record sequence in the original database, and count the support of all the candidate k -large-sequences in all the k connection records. Suppose there are n connection records in database, the number of k connection record will be C_n^k . If we dig out a n large sequence, the number of connection record will be $C_n^1 + C_n^2 + \dots + C_n^n = 2^n$. If we mine an attack with 100 connection records, the structure is not suitable for sequential pattern mining.

When we search for L_1 , our algorithm is the same as Apriori algorithm. When searching for L_2 , we only need to scan the temporary database which is composed of L_1 instead of original database. The data quantity reduces evidently. And the most important, we take out a C_2 , and only scan the two L_1 s which may compose the C_2 in temporary database. Then, take out other C_2 in turn. When we search for L_k , we only need to scan the temporary database which is composed of L_{k-1} . We take out a C_k , and only scan the two L_{k-1} s which may compose the C_k in temporary database. And then, take out other C_k in order. Now, we get $SL_1: L_1, L_2, \dots, L_k$. When searching for SL_k , we only need to scan the temporary database combined by SL_{k-1} , and only scan the two SL_{k-1} s which may compose the SC_k in temporary database. The complexity of searching for SL_k is the same as searching for L_k .

Thus, compared with AprioriAll algorithm, our algorithm is proved efficiency, especially when the number of connection records is relatively more, and the number of item is relatively less.

3.4 Sequential Pattern Comparison and Match

We dig out a large sequence tree from intrusion and normal dataset with sequential pattern mining algorithm. In order to obtain "intrusion-only" large sequence tree, we also need to subtract the same section of two trees from the intrusion large sequence tree. It is difficult to get the same section of the two trees directly. We can match the intrusion large sequence tree with multi-instances of normal dataset. And then we mark the matched nodes. If the number of the marked nodes is greater than the given minimal support, we know that the nodes should not in the "intrusion-only" tree. So, we get an "intrusion-only" tree with mark. When we match ready detection data, the marked nodes are not output in the sequence of intrusion detection model.

Suppose that the zero large sequence in the "intrusion-only" large sequence tree is an empty root node. We link all its child nodes into a candidate chain in the order. We

take out the first ready detection record, and match each attribute and its combination to the first node of the candidate chain. If it does not match, we match the second node along the candidate chain. If it matches, we mark the node, and link all its child nodes into a new chain to replace the node, and then we match the second node along the candidate chain, till the last node of the candidate chain. Then, we get a new candidate chain. We take out the second ready detection record, and match the new node as the way, till the last ready detection record. If a leaf node of “intrusion-only” large sequence tree is marked, the sequence means that the intrusion occurs.

4 Experiment Result and Analysis

We dig out a sequential pattern of casesen attack with the misuse intrusion detection algorithm based on sequential pattern, which is described in figure 6.

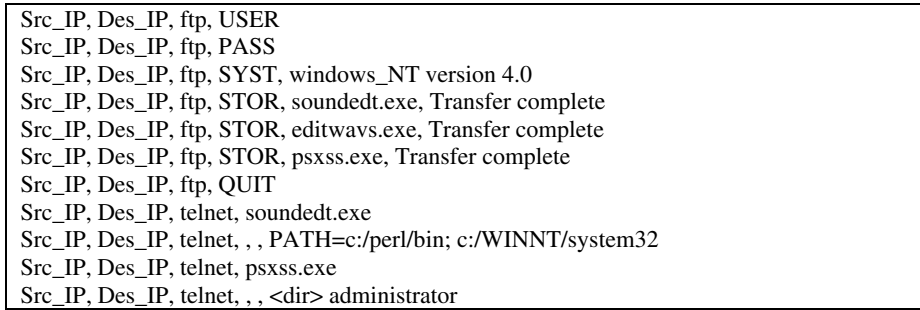


Fig. 6. A sequential pattern of casesen attack

Figure 6 is a casesen attack, which belongs to U2R attack for windows NT. This is the longest large sequence of the attack, its subsequences can be acted as fuzzy match set. The attacker ftps three attack files to the victim: soundedt.exe, editwavs.exe, psxss.exe. Then, the attacker telnets to the victim and runs soundedt.exe. A new object is created in the NT object directory which links to the directory containing the attack files. A posix application is started activating the trojan attack file, psxss.exe, which results in the logged in user being added to the Administrators user group. At last, the attacker removes the intrusion trail.

Casesen attack has distinct time order, but not distinct character string features. Most features appear only once in an attack, which are difficult to take statistic. The method of character string matching in a packet and sensitive information Statistics cannot describe the attack exactly. We need to combine multi-features in time order.

In addition, not only the longest sequence can represent the appearance of an attack. When the case that some packets miss, if the length of matching sequence in the sequence tree is longer than the threshold or the number of matching sequences is more than the threshold, the attack can be proved occurrence too. This is called fuzzy matching,

Sequential pattern mining algorithm is based on other intrusion detection algorithm. By preprocessing data, we can extract sensitive information from transmit data

with character string matching as an attribute of sequential pattern mining algorithm. We can also classify the sensitive information and perform statistical analysis.

The experiment indicates that the sequential pattern we obtained is different from command sequential pattern, it is a new sequential pattern. It can describe attacks more accurately, and detect the attacks whose features appear only once, and improve detection rate, and offer a new idea for the research of intrusion detection.

5 Conclusion

How to generate new training data is the intrinsic disadvantage of machine learning. Except for downloading standard training data from Internet, we can build attack-defense simulation plate, and apply new attacks to the simulation system, collect the generating data as training data, which offers a new approach for the research of misuse intrusion detection. We can improve our method by:

- Perfecting protocol analysis tools, parsing the information of different protocol layer as data source of data mining;
- Optimizing sequential pattern mining algorithm, simplifying sequential pattern, resolving fuzzy match problem;
- Combining pattern, and building less state machine models, when building multi-intrusion models.

References

1. G. R. Malan and F. Jahanian. An extensible probe architecture for network protocol performance measurement. In *Proceedings of SIGCOMM*, Vancouver, British Columbia, Canada, 1998, 215 – 227.
2. VernPaxon. Bro: A system for detecting network intruders in real-time. *Computer Networks*, 1999, 31(23-24), 2435-2463.
3. W.Lee. A Data Mining Framework for Constructing Feature and Model for Intrusion Detection System. [D]In the Graduate School of Arts and Sciences, COLUMBIA UNIVERSITY. 1999.
4. LIAN Yi-Feng, Research of Distributing intrusion detection system, [D]In University of Science and Technology of China , China, May 2002.
5. S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. [J]In *Journal of Computer Security*, 1998, 6:151–180.
6. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *SIGKDD*, 2002, p429–435.
7. DARPA Intrusion Detection Evaluation <http://www.ll.mit.edu/IST/ideval/>.
8. WEN Yan, The Research and Implementation on The Network Behavior Monitor System Based on The General Finite States Machine, [D] In Paper of the degree of Master in School of Computer Science, National University of Defense Technology, China, May 2004.

Network Performance Measurement Methodologies in PGMS

Yuanzhe Yao, Binxing Fang, Hongli Zhang, and Wei Wang

Research Center of Computer Network and Information Security Technology
Harbin Institute of Technology, Harbin, Heilongjiang, P.R. China, 150001
yyz@pact518.hit.edu.cn

Abstract. PGMS (P2P-based Grid Monitoring System) is a Grid monitoring system, which is based on peer-to-peer technology and GMA specification. Network performance measurement is a key component of monitoring system. Always, by all means, many monitoring systems avoid intrusiveness introduced by measurement and interferences between multiple sensors. In PGMS, the basic architecture and function cell is the peer, and some peers constitute a peer group, which is the management cell of PGMS and interconnected by peer-to-peer means. The architecture character of PGMS imposes special influences on network performance measurement. Some network measurement methodologies and rough estimate algorithm of bandwidth and delay are presented, which are based PGMS framework. Adaptive control of sensors and on-demand measurement methods are adopted to reduce intrusiveness and interferences, and increase scalability of PGMS.

1 Introduction

The aim of Grid is to become a high performance, high throughput, and high productivity computing platform. The technology developments of distributed computing, supercomputing, Internet and high speed networks have made it possible. At present, even the rapid development of high performance computer can't satisfy the sharp increase of amount of calculation in many fields. It may demand unprecedented computation power to solve some huge problem of scientific, engineering, and commercial field. The only answer to the problem is to interconnect various heterogeneous computing resources distributed in global locations to a virtual super computer. Grid gathers usable computation power or resources and then ubiquitously and seamlessly provides manifold Grid services to user. Grid system is an extremely complicated distributed computing environment. In Grid, the kind and number of resources is changeful, and the performances of resources are also fluctuating. Therefore, it is very important to efficiently monitor and manage resources and applications running on Grid.

The success of Internet and development of network technology is an important element of Grid progress. Network monitoring and measurement is an important function unit of a well-designed Grid monitoring system. It should be able to measure

the performance and status information of network, and these information are used widely, such as fault detection, debugging, performance analysis and tuning, load balancing, task scheduling, security, auditing and accounting.

In this paper, Section 2 introduces the meanings of network measurement methodology and an actual example: NWS. In section 3, we present some network measurement methods and rough estimate algorithm of bandwidth and delay that are based PGMS framework, and then the research on adaptive control of sensors and on-demand measurement methods are done in order to reduce intrusiveness and interferences. In the last section, we present future research work.

2 Measurement Methodology

2.1 Meanings of Network Measurement Methodology

At present, there are very many network measurement tools and system [1] [2]. In them, all network characteristics may have been taken into account. In Grid, GGF Network Measurements Working Group (NMWG) [3] has defined some characteristics of network. [4]

Even though the same characteristic, such as delay, there may be different measurement method, for example, one-way delay may directly be measured between two sites with precise time synchronization, also, may be acquired by the RTT indirectly and approximately.

Although, there are many measurement tools and systems for LAN or WAN, only a few of them are suit for Grid monitoring. An important reason is that the management mechanism and running method of sensors are not satisfactory.

Measurement methodology can impose very important effect on measurement results and efficiency. It refers to specific measurement methods and measurement policy. Here, we mainly consider the latter in this paper. Later, we use an example to show the importance and representation form of measurement methodology.

2.2 An Example: NWS

The NWS (Network Weather Service) is a distributed monitoring and forecasting system that operates a set of performance sensors such as network sensors, CPU sensors. From these sensors the NWS gathers readings of the instantaneous status and uses one of multiple numerical models to generate forecasts of what the conditions will be for a given time frame. Since this prediction functionality is analogous to weather forecasting, the system is named as Network Weather Service [5].

The NWS may periodically monitor and dynamically predict network and computing resources performance. The NWS mainly measures network performance and CPU availability, and then employs the data to predict these performances. NWS design is simple and comprehensive. This is a tenet of the NWS. It is expected that any similar monitoring tool apply the same methods. The NWS was designed as a modular system to provide performance information for distributed application

scheduling. The NWS must be able to sense the performance of resources in whole system, predict the future performance and hand out the forecasting information to the user.

In order to make the NWS sensors report available resources performance observed by sensors, each monitored host runs a copy of the NWS server. Each server maintains a network performance sensor, a CPU loading sensor, and a memory sensor. All of the servers in the system share a public host list and a TCP port number list. Every server periodically select a host from the host list and carry out a communication test with it, and then the latency, throughput, and effective throughput are recorded in an internal database.

In Grid monitoring system, multiple sensors may bring some problems. When measuring is taken, it has to consume a part of resources and then inevitably brings error itself, that is, "To measure how much we have, we must consume a part of what we measure" [6]. This is the intrusiveness problem of measuring. When a system has more than one probe or sensor, they may influence on each other. This is the interference problem. For example, the NWS server works on a logical, application level network topology, and does not know which linkage is the medium that can be exclusively accessed (for example, the Ethernet), and which is not. If communication test comply with the local clock, at the same time two or more servers may select the same link to make a test. Thus, interference occurs.

Therefore, in order to provide precise predictions, the measurements must be not so intrusive as possible. Performance experiments that measure the deliverable performance at any given time must also not interfere with each other. Otherwise, the data with errors will be brought in to the generated forecasts.

To solve interference problem, when the NWS sensors make a measure, an external management process is needed to make synchronization. Concretely, it solves the problem using a token passed between the servers; the result is that the one getting the token has the authority to test. [7]

3 Network Measurement Methodologies in PGMS

3.1 P2P-Based Grid Monitoring System

P2P-based Grid Monitoring System (PGMS), based on peer-to-peer technology and GMA architecture specification, is designed to monitor large scale and dynamically changed Grid system. In PGMS, the function of directory service is achieved by the P2P-based Grid Distributed Directory Service (PGDDS), in which a whole directory service is decentralized into some associated directory services that are peer-to-peer. PGDDS are not only independent each other but also related nearly. In this way, they cooperate to implement a single directory image.

In PGMS, each Grid node is called as a *peer*, which runs a copy of PGMS program. Each copy consists of instrumentation sensor, instrumentation management and control module, communicating and publication module, data preprocessing module, data caches and archive module, performance forecasting module and so on. Instrumentation sensors include host sensor, network sensor and software sensor.

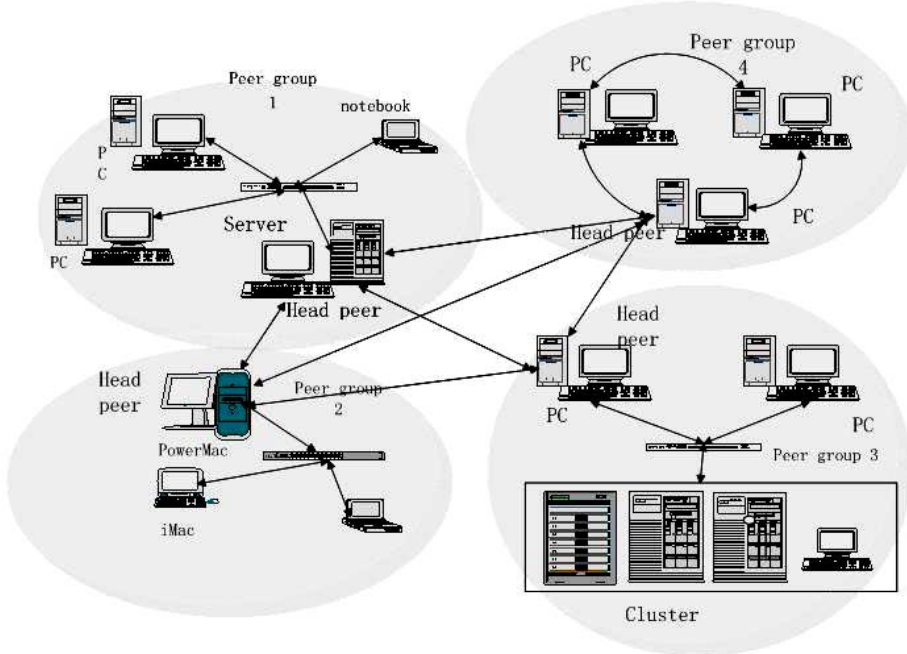


Fig. 1. A deployment sketch map of PGMS

Network performance measurement is a key application of monitoring. The aims of network measurement are to obtain elementary data of interesting peers, such as bandwidth, delay, routing, throughput, and so on. These data are vital to task scheduling, debugging, data division, program performance analysis and tuning, traffic engineering.

As figure 1 showed, some related peers constitute a *peer group* by some autonomous means. In every peer group, there is a *head peer* and a *backup peer* that is the backup of head peer. Head peer is elected to play a core and key role in PGMS. A PGDDS consists of two components: a global information publication directory (GIPD), which inhabits in head peer, provides the information of other head peers in other peer groups, and a local information publication directory (LIPD), which provides the information of peer in a same group, is maintained by each peer including head and backup peer. Backup peer is appointed by head peer and owns a same GIPD and LIPD with head peer. Periodically, head peer updates the GIPD and LIPD information. Head peer has the authority to manage the other peers in the same peer group.

Universal Resource Description Word (URDW) is a resource description method. It is distinct from general methods that use some resource description language, such as XML. A URDW can include all key information of a peer, so while the other peers obtain a URDW, they can decode it and know the running states of the peer.

3.2 Adaptive Control of Network Sensors and On-Demand Network Measurement in PGMS

In PGMS, the basic architecture and function cell is the peer, and some peers constitute a peer group, which is the management cell of PGMS and interconnected by peer-to-peer means. The architecture character of PGMS imposes special influences on network performance measurement, and all network measurements should embody and take advantage of these characters.

There are two kinds of measure means: active means and passive means, it is major distinction that the former impose certain load on the measured objects, then observe and record results, the latter don't add loads instead. Passive measurement is performed by observing network traffic, and does not disturb the network. It is mainly used to measure traffic flows. Active measurement, on the other hand, imposes extra traffic onto a network and can disturb its behavior, thereby affecting measurement results. Network measurement often use the active means, as a result, it is inevitable to introduce intrusiveness and interference between the probes. Always, many monitoring systems avoid intrusiveness introduced by measurement and interferences between multiple sensors by all means. Therefore, it is necessary to correctly control the behavior of sensors.

At present, there are many measurement tools and systems for LAN or WAN, but only a few of them are suit for Grid monitoring. An important reason is that the management mechanism and running method of sensors are not satisfactory. In PGMS, IMC module manages the all sensors in peer. Although a peer has the same sensors with another peer does, it should be entirely different that how to operate these sensors according to the differences of condition and context.

Mainly, the network sensors control focus on the deployment and running policy of network sensors. Concretely, the contents of control include that where are the sensors placed, when are they invoked, how to cooperate with each other, and the frequency and order of their running.

For example, sensors operate may not be periodical, and changeable frequency may more suit for some applications, when the measured network character change very slowly, the frequency should be low, and, contrariwise, the frequency should be high.

Moreover, the contents, frequency and mode of measurement are various with the different application aims. Network topology discovery may mainly be concerned about the routing information; the task scheduling is sensitive to end-to-end bandwidth and delay between the peers participating in computing and throughput of peers. As the input of network performance forecasting and the basis of prediction error calculating, the measurement data precision will directly influence feasibility and reliability of performance forecasting module. Additionally, the change of network traffic, the kind and number of data package of are an important source of network security status monitoring.

In Grid, it is impracticable to make end-to-end measurement between all peers. Moreover, even the measurements between parts of them are not practical because of huge overheads. Therefore, basing on the well-controlled measuring and probing, on-

demand measurement is a good solution, which can reduce intrusiveness and increase scalability of PGMS. Here, on-demand network measurement means that measurement operates between the special peers on the time demanded by the consumers of measurement event data, in this time, network sensors in operation are the producers of measurement event data.

3.3 Rough Estimate of Network Bandwidth and Delay

Here, the rough estimate algorithm means the data produced by the algorithm is not precise enough to act as the input of key applications, but the precision of them is sufficient for ordinary applications, which need not very high precision. Moreover, these data may not be final, and when the precise data are needed, a new measurement operation takes place, which may be consistent with the rough estimate to a great extent.

In PGMS, assuming there exist N peer groups: $PG_1, PG_2, \dots$ and PG_N , and their head peers are respectively $HP_1, HP_2, \dots$ and HP_N . Let P_{mn} is the n th peer of the PG_m , $m < N$, B_{ij} is the bandwidth between the HP_i and HP_j , $i, j < N$, IB_{ij} is the bandwidth between the HP_i and P_{ij} , $B_{mn,lk}$ denotes the bandwidth between P_{mn} and P_{lk} in PGMS. D_{ij} is the delay between the HP_i and HP_j , $i, j < N$, ID_{ij} is the bandwidth between the HP_i and P_{ij} , $D_{mn,lk}$ denotes the bandwidth between P_{mn} and P_{lk} in PGMS, then

Algorithm 1:

$$B_{mn,lk} = \min \{IB_{mn}, B_{ml}, IB_{lk}\} \quad m \neq l \quad (1)$$

$$B_{mn,lk} = B_{ml}, \text{ when } IB_{mn} > B_{ml}, \text{ and } IB_{lk} > B_{ml} \quad m \neq l \quad (2)$$

Algorithm 2:

$$D_{mn,lk} = ID_{mn} + D_{ml} + ID_{lk} \quad m \neq l \quad (3)$$

$$D_{mn,lk} = D_{ml}, \text{ when } ID_{mn} \ll D_{ml}, \text{ and } ID_{lk} \ll D_{ml} \quad m \neq l \quad (4)$$

Formula 1 and formula 3 approximately give rough estimate value of network bandwidth and delay between the arbitrary two peers which are not in the same peer group in PGMS. Further, when the inner bandwidth in peer groups greater than the one between the peer groups, the bandwidth between a arbitrary pair of peers, which are not in same peer group, approximate to the one between two head peers (formula 2); and when the inner delays in peer groups are far less than those between head peers, delay between a arbitrary pair of peers, which are not in same peer group, approximate to the one between two head peers (formula 4). The situation may come forth when some peer groups are in same LAN and they are interconnected by WAN.

When the peers are in the same peer group, the bandwidth and delay should be measured directly. They need not and should not be computed indirectly, because the value acquired by computing may be too rough to use.

3.4 An Example: On-Demand Measurement in Task Scheduling

The network sensors should have different running method from the other sensors in that their measurement operations refer to multiple peers in network, and the other sensors only influence on the host peer. So host sensors and application sensors may update measurement data more frequently.

Using algorithms in section 3.3, we may obtain an estimate of network bandwidth and delay between all peers in PGMS by simple calculating. However, these data have not enough precision to act as inputs of key applications, for example, task scheduling. Here, we use scheduling as example to illustrate how the on-demand measurement works.

As a task scheduler, primarily, it should evaluate the task waiting for scheduling in order to know the resource demand of the task, which may include the number, frequency, available of CPU, the capacity, available of memory, and the bandwidth and delay of network, and so on. These data are divided into two sections: the data about network performance and those about the other resources. These data are converted into a URDW, and then the scheduler submits the URDW to PGDDS. Where some candidate peers are found, which are coincident to requirements described by the URDW, but not all the peers are able to become final electees because their resources data are rough estimate value.

As we know, accordingly, the network performance data of the candidate peers need be remeasured to acquire enough precision. In this way, basing on the remeasured data and historical data, from the candidate peers, the scheduler selects most appropriate peers as final electees and assign corresponding task to them.

Obviously, the on-demand measurement is a partial measurement, and has prominent pertinence. Therefore, the overheads also are far smaller than the full-scale methods. Certainly, if rough estimate value is enough precise, the remeasurement is not necessary and nonsense because it introduce more overhead and intrusiveness.

4 Conclusions and Future Work

Measurement methodology can impose very important effect on measurement results and efficiency. It is essential that different measurement methodology should be adopted according to conditions and contexts. The approximate estimate on bandwidth and delay, and on-demanded measurement are suit for the framework of PGMS and helpful to reduce overheads and intrusiveness, and avoid interference.

Currently, the PGMS is being implemented. During the procedure of implementation, some original ideas may be testified to be true, and others need to be modified, but the others are disconfirmed. Nevertheless, the exploring to network measurement methodology is significative to some extent, and should continue in future.

References

1. <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>
2. <http://www.caida.org/tools/taxonomy/performance.xml>

3. <http://www-didc.lbl.gov/NMWG/>
4. A Hierarchy of Network Performance Characteristics for Grid Applications and Services, <http://www-didc.lbl.gov/NMWG/docs/draft-ggf-nmwg-hierarchy-03.pdf>
5. Rich. Wolski, Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1998, (1): 119-132.
6. <http://www.pagebox.net/nws.html>
7. Benjamin Gaidioz, Rich Wolski, Bernard Touran cheau, Synchronizing network probes to avoid measurement intrusiveness with the Network Weather Service. *High-Performance Distributed Computing*, 2000. Proceedings. The Ninth International Symposium on, 1-4 Aug. 2000: 147 –154.

Via Firewalls

Lu Yan

Turku Centre for Computer Science (TUCS) and
Department of Computer Science, Åbo Akademi University
FIN-20520 Turku, Finland
`Lu.Yan@abo.fi`

Abstract. A lot of networks today are behind firewalls. In peer-to-peer networking, firewall-protected peers may have to communicate with peers outside the firewall. This paper shows how to design peer-to-peer systems to work with different kinds of firewalls within the object-oriented action systems framework by combining formal and informal methods. We present our approach via a case study of extending a Gnutella-like peer-to-peer system to provide connectivity through firewalls.

1 Introduction

The idea of peer-to-peer networking, in a sense that nodes on the network communicate directly with each other, is as old as Internet itself. Internet used to be a peer-to-peer network if we go back to those early days in the 70's when Internet was limited to researchers in a few selected laboratories. Nowadays Internet has developed into a non peer-to-peer network, in the sense that most exchanges rely on mediation through gateways and servers. Moreover, most networks today employ firewalls, for security reasons, which impede direct communication by filtering packets and limiting the port numbers open to bi-directional traffic.

The vision of peer-to-peer networking is to remove the distinction between client and server. Instead of running web browsers that can only request information from web server, users can run peer-to-peer applications to contribute contents or resources in addition to requesting them. As a vision of peer-to-peer networking, it is necessary for peer-to-peer applications to work in most environments, whether home, small business, or enterprise.

Previously [1], we have specified a Gnutella-like peer-to-peer system within the object-oriented action systems [2] framework by combining UML diagrams. When implementing such a system in Java, we realized that a lot of networks today are behind firewalls. In peer-to-peer networking, firewall-protected peers may have to communicate with peers outside the firewall. Thus a solution should be made to create communication schemes that overcome the obstacles placed by the firewalls to provide universal connectivity throughout the network. This motivates us to conduct a study of firewalls in peer-to-peer networking and achieve a way to traverse firewalls. We present our approach via a case study of extending a Gnutella-like peer-to-peer system to provide connectivity through firewalls.

2 Uni-directional Firewalls

Most corporate networks today are configured to allow outbound connections (from the firewall protected network to Internet), but deny inbound connections (from Internet to the firewall protected network) as illustrated in Fig. 1.

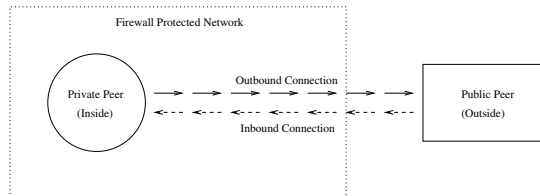


Fig. 1. Uni-directional Firewall

These corporate firewalls examine the packets of information sent at the transport level to determine whether a particular packet should be blocked. Each packet is either forwarded or blocked based on a set of rules defined by the firewall administrator. With packet-filtering rules, firewalls can easily track the direction in which a TCP connection is initiated. The first packets of the TCP three-way handshake are uniquely identified by the flags they contain, and firewall rules can use this information to ensure that certain connections are initiated in only one direction. A common configuration for these firewalls is to allow all connections initiated by computers inside the firewall, and restrict all connections for computers outside the firewall. For example, firewall rules might specify that users can browse from their computers to a web server on Internet, but an outside user on Internet cannot browse to the protected user's computer.

In order to traverse this kind of firewalls, we introduce a new descriptor and routing rules for servants [3].

Push. A mechanism that allows a firewalled servant to contribute file-based data to the network. A servant may send a Push descriptor if it receives a QueryHit descriptor from a servant that doesn't support incoming connections.

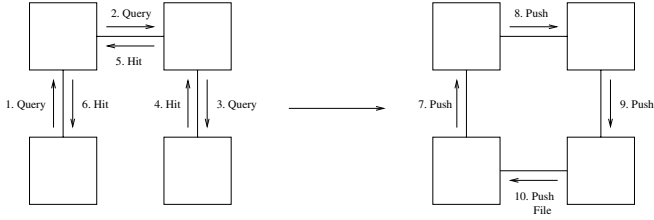
The message format [1] has to be revised to adopt the new descriptor. The message *type* now includes *Ping*, *Pong*, *Query*, *QueryHit* and *Push*, so minor changes are made in Table 1.

Once a servant receives a QueryHit descriptor, it may initiate a direct download, but it is impossible to establish the direct connection if the servant is behind a firewall that does not permit incoming connections to its Gnutella port. If this direct connection cannot be established, the servant attempting the file download may request that the servant sharing the file *Push* the file instead. i.e. A servant may send a Push descriptor if it receives a QueryHit descriptor from a servant that doesn't support incoming connections.

Unlike the previous descriptors Ping, Pong, Query and QueryHit, Push descriptors are routed by *ServentID*, not by *DescriptorID*. Intuitively, Push descrip-

Table 1. Message format
$$\begin{aligned}
Msg = & \ll \textbf{attr} \quad descriptorID; serventID; TTL; type; info \\
& \textbf{meth} \quad Transmit() = TTL > 0 \rightarrow TTL := TTL - 1 \\
& \textbf{init} \quad t \in \{Ping, Pong, Query, QueryHit, Push\} \rightarrow \\
& \quad \quad \quad Msg(t) = (descriptorID := \underline{unique\_ID}; \\
& \quad \quad \quad TTL := \underline{max\_TTL}; type := t; info := \underline{info\_}) \\
& \rr
\end{aligned}$$

tors may only be sent along the same path that carried the incoming QueryHit descriptors as illustrated in Fig. 2. This ensures that only those servents that routed the QueryHit descriptors will see the Push descriptor. A servent that receives a Push descriptor with $ServentID = n$, but has not seen a QueryHit descriptor with $ServentID = n$ should remove the Push descriptor from the network.

**Fig. 2.** Push routing [3]

We adopt uni-directional firewalls by adding a Push router in Table 2, which is an action system Rf modeling Push routing rules. Since this action system actually models a particular aspect of a full router, we can compose it with the previous two action systems Rc modeling Ping - Pong routing rules and Rl modeling Query - QueryHit routing rules together, using *prioritizing composition* [4] to derive a new action system specification of a full router

$$R = \ll Rc \parallel Rl \parallel Rf \rr$$

where on the higher level, we have components of the router

$$\begin{aligned}
& \{ \langle Router, R \rangle, \langle PingPongRouter, Rc \rangle, \\
& \langle QueryRouter, Rl \rangle, \langle PushRouter, Rf \rangle \}
\end{aligned}$$

A servent can request a file push by routing a Push request back to the servent that sent the QueryHit descriptor describing the target file. The servent that is the target of the Push request should, upon receipt of the Push descriptor, attempt to establish a new TCP/IP connection to the requesting servent. As specified in the refined file repository in Table 3, when the direct connection is

Table 2. Specification of Push router

```

Rf = [| attr  serventDB :=  $\phi$ ; cKeyword :=  $\phi$ ; filename :=  $\phi$ ;
           target :=  $\phi$ ; pushTarget :=  $\phi$ ;
obj  receivedMsg : Msg; newMsg : Msg; f : FileRepository
meth SendPush( ) = (newMsg := new(Msg(Push)));
           newMsg.info.requestIP := this_IP;
           newMsg.info.filename := receivedMsg.info.filename;
           newMsg.info.destinationIP := receivedMsg.info.IP;
           _outgoing_message_ := newMsg);
           ReceiveMsg( ) = receivedMsg := _incoming_message_;
           ForwardMsg(m) = (m.TTL > 0  $\rightarrow$ 
           m.Transmit( ); _outgoing_message_ := m)

do
  true  $\rightarrow$ 
    ReceiveMsg( );
    if receivedMsg.type = QueryHit  $\rightarrow$ 
      serventDB := serventDB  $\cup$  receivedMsg.serventID;
      if receivedMsg.info.keyword = cKeyword  $\rightarrow$ 
        target := receivedMsg.info.filename@
        receivedMsg.info.IP;
        if f.firewall  $\rightarrow$ 
          SendPush( )
        fi
        cKeyword :=  $\phi$ 
      [| receivedMsg.info.keyword  $\neq$  cKeyword  $\wedge$ 
        receivedMsg.descriptorID  $\in$  descriptorDB  $\rightarrow$ 
        ForwardMsg(receivedMsg)
      fi
    [| receivedMsg.type = Push  $\rightarrow$ 
      if receivedMsg.info.destinationIP = this_IP  $\rightarrow$ 
        pushTarget := receivedMsg.info.requestIP@
        receivedMsg.info.filename@
        receivedMsg.info.destinationIP
      [| receivedMsg.info.destinationIP  $\neq$  this_IP  $\wedge$ 
        receivedMsg.serventID  $\in$  serventDB  $\rightarrow$ 
        ForwardMsg(receivedMsg)
      fi
    fi
od
|]

```

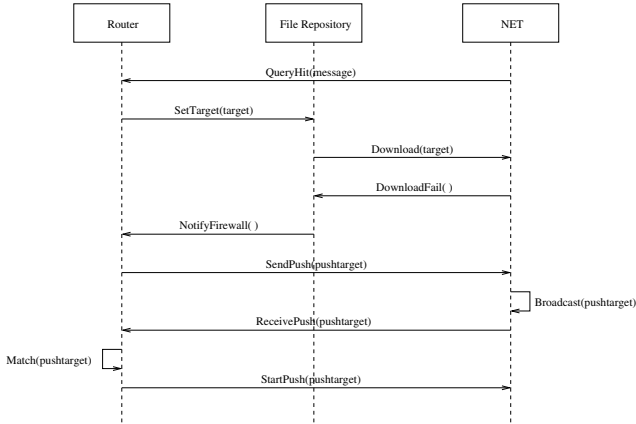
established, the firewalled servent should immediately send a HTTP GIV request with *requestIP*, *filename* and *destinationIP* information, where *requestIP* and *destinationIP* are IP address information of the firewalled servent and the target servent for the Push request, and *filename* is the requested file information. In this way, the initial TCP/IP connection becomes an outbound one, which is allowed by uni-directional firewalls. Receiving the HTTP GIV request, the target

Table 3. Specification of file repository

```

F = [[ attr  firewall* := false; fileDB := _fileDB_; cFileDB;
        filename :=  $\phi$ ; target :=  $\phi$ ; pushTarget :=  $\phi$ 
    meth  SetTarget(t) = (target := t);
        PushTarget(t) = (pushTarget := t);
        Has(key) = ( $\{key\} \in \text{dom}(fileDB)$ );
        Find(key) = (filename := file  $\wedge$   $\{file\} \in \text{ran}(\{key\} \triangleleft fileDB)$ )
    do
        target  $\neq \phi \rightarrow$ 
            cFileDB := fileDB;
            HTTP_GET(target);
            target :=  $\phi$ ;
            Refresh(fileDB);
            if fileDB = cFileDB  $\rightarrow$ 
                firewall := true
            [] fileDB  $\neq$  cFileDB  $\rightarrow$ 
                firewall := false
            fi
            [] pushTarget  $\neq \phi \rightarrow$ 
                HTTP_GIV(pushTarget);
                pushTarget :=  $\phi$ ;
                Refresh(fileDB)
    od
]]

```

**Fig. 3.** Sequence diagram of a Push session

servent should extract the *requestIP* and *filename* information and construct an HTTP GET request with the above information. After that, the file download process is identical to the normal file download process without firewalls. We summarize the sequence of a Push session in Fig. 3.

3 Port-Blocking Firewalls

In corporate networks, another kind of common firewalls are port-blocking firewalls, which usually do not grant long-time and trusted privileges to ports and protocols other than port 80 and HTTP/HTTPS. For example, port 21 (standard FTP access) and port 23 (standard Telnet access) are usually blocked and applications are denied network traffic through these ports. In this case, HTTP (port 80) has become the only entry mechanism to the corporate network. Using HTTP protocol, for a servent to communicate with another servent through port-blocking firewalls, the servent has to *pretend* that it is an HTTP server, serving WWW documents. In other words, it is going to mimic an *httpd* program.

When it is impossible to establish an IP connection through a firewall, two servents that need to talk directly to each other, solve this problem by having SOCKS support built into them, and having SOCKS proxy running on both sides. As illustrated in Fig. 4, it builds an HTTP-tunnel between the two servents.

After initialization, the SOCKS proxy creates a *ProxySocket* and starts accepting connections on the Gnutella port. All the information to be sent by the attempting servent is formatted as a URL message (using the GET method of HTTP) and a *URLConnection* via HTTP protocol (port 80) is made. On the other side, the target servent accepts the request and a connection is establish with the attempting servent (actually with the SOCKS proxy in the target servent). The SOCKS proxy in the target servent can read the information sent by the attempting servent and write back to it. In this way, transactions between two servents are enabled.

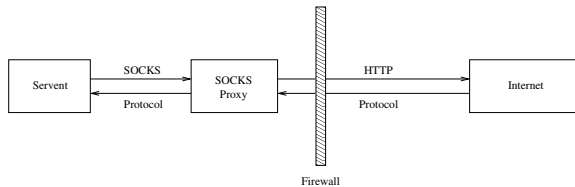


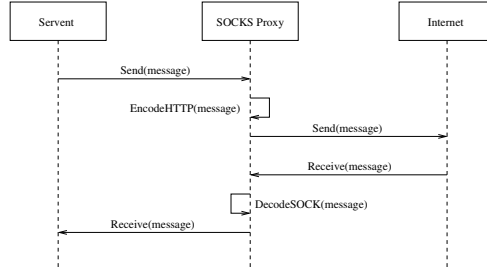
Fig. 4. Firewall architecture and extendable socket

We adopt port-blocking firewalls by adding a *SOCKS proxy* layer to the architecture of servent. This layer will act as a tunnel between servent and internet. As specified in Table 4, after receiving messages from the attempting servent and encoding them into HTTP format, the SOCKS proxy sends the messages to internet via port 80. In the reverse way, the SOCKS proxy keeps receiving messages from HTTP port and decoding them into original format. With this additional layer, our system can traverse port-blocking firewalls without any changes in its core parts. We summarize the sequence of a SOCKS proxy session in Fig. 5.

Table 4. Specification of SOCKS proxy

```

S = [[ attr listenPort := Gnutella_port;
        destinationPort := 80
      obj ProxySocket : Socket;
        HTTPSocket : Socket;
        imsg : Msg; omsg : Msg
      init ProxySocket = new(Socket(listenPort));
        HTTPSocket = new(Socket(destinationPort))
      do
        _incoming_request_ ≠  $\phi$  →
          imsg := EncodeSOCK(DecodeHTTP(HTTPSocket.Read()));
          _incoming_message_ := ProxySocket.Write(imsg)
        || _outgoing_request_ ≠  $\phi$  →
          omsg := EncodeHTTP(DecodeSOCK(ProxySocket.Read()));
          _outgoing_message_ := HTTPSocket.Write(omsg)
      od
    ]]
```

**Fig. 5.** Sequence diagram of a SOCKS proxy session

4 Related Work and Concluding Remarks

There have been protocols such as PPTP (Point-to-Point Tunneling Protocol), UPNP (Universal Plug and Play), RSIP (Realm Specific IP) and Middlebox protocol to address the firewall problems in peer-to-peer networking. A recent protocol, JXTA [5] has provided an alternative solution to the firewall problem by adding a publicly addressable node, called “rendezvous server”, which firewalled peer can already talk to. The scheme is that peers interact mostly with their neighbors who are on the same side of the firewall as they are and one or a small number of designated peers can bridge between peers on the different sides of the firewall. But the problem posed by firewalls still remains when configuring the firewalls to allow traffic through these bridge peers.

In this paper, we have presented our solution to traverse firewalls for peer-to-peer systems. We have extended a Gnutella-like peer-to-peer system to adopt uni-directional firewalls and port-blocking firewalls using OO-action systems.

During the extending work, our experiences show that the object-oriented aspect of OO-action systems helps to build systems with a reusable, composable and extendable architecture. The modular architecture of our system makes it easy to incorporate new services and functionalities without great changes to its original design.

Peer-to-peer networking is currently attracting lots of attention, spurred by the surprisingly rapid deployment of some peer-to-peer applications like BitTorrent, Kazaa and eMule. Firewalls have become a great challenge to peer-to-peer networking upon Internet. In the future work, we plan to explore more sophisticated protocols like SOAP [6] and incorporate them into the development of peer-to-peer systems to provide safe and reliable access via firewalls.

References

1. L. Yan and K. Sere: *Stepwise Development of Peer-to-Peer Systems*. Proceedings of the 6th International Workshop in Formal Methods (IWFM'03), Dublin, Ireland, July 2003. Electronic Workshops in Computing (eWiC), British Computer Society Press.
2. R.J.R. Back and K. Sere: *From Action Systems to Modular Systems*. Software - Concepts and Tools. (1996) 17: 26-39.
3. Clip2 DSS: *Gnutella Protocol Specification v0.4*.
Online. <http://www.clip2.com/GnutellaProtocol04.pdf>.
4. E. Sekerinski and K. Sere: *A Theory of Prioritizing Composition*. The Computer Journal. Vol. 39, No.8, 1996.
5. L. Gong: *JXTA: A network programming environment*. IEEE Internet Computing, 5(3): 88-95, May/June 2001.
6. W3C: *Simple Object Access Protocol (SOAP)*.
Online. <http://www.w3c.org>.

A Grid Security Infrastructure Based on Behaviors and Trusts*

Xiaolin Gui, Bing Xie, Yinan Li, and Depei Qian

Department of Computer Science and Technology, Xi'an Jiaotong University
710049, Xi'an China
xlgui@mail.xjtu.edu.cn, xiexiebing@sohu.com

Abstract. Computational Grids need support the distributed high performance computing with security and reliability. While the fact is when malicious users apply computational resources or illegally modify their secure levels, they can get system-level data or outputs belong to other applications by running cock-horse, even more they can destroy the whole system. For solving these security problems in Grids, a Grid security model with users behaviors and trusts is introduced. And by improving the components of reputation in traditional trust [1], [2] model, a new trust model with mathematical description is presented, and the grid security infrastructure with this trust model is described.

1 Introduction

The purpose of developing Grid is to aggregate resources from Internet for high-performance computing and wide-area information services. Grid can be seen as a super virtual supercomputer, it supports general resources sharing, including computational resources, memory resources, data resources and costly machines, etc. And from the above sharing, Grid users can get cooperative high performance computing services and wide-area information services. In computational Grids, for implementing secure and reliable high performance computing service, the study on how to support security infrastructure for Grid is necessary.

Although most popular computational Grids and their toolboxes both include certain secure techniques, and security of resources and applications is still a challenge. Security Management Mechanism in Globus consists of GSI [3] and GSS-API. GSI mainly points to secure the transport layer and application layer in network, and emphasizes on introducing present popular security techniques into Grids environment. Legion [4] implements security mechanism by applying the oriented objects. In Legion domains, every object owns different secure levels, and simultaneously, these secure levels can be added or reduced freely.

Now, network security techniques are developing more and more mature, but there are many kinds of restrictions when they are put into Grids. For example, systems

* This work was sponsored by the National Science Foundation of China (60273085), the state High-tech Research and Development project (Grid Software) and the ChinaGrid project of China.

usually require the resources in different management domains can be trusted by each other, users must be all legal, applications are totally harmless, etc. These restrictions deeply frustrate the scale of Grids, users and applications. At first, while actual computational Grids can span several management domains, and in these domains, the very high trust relationship must be supported in each other. Secondly, if illegal users run cockhorse in Grid, certain resources may be destroyed and all information in it will disappear forever.

For solving the above problems, this paper study on how to implement the management of trust level supporting assigning and updating dynamically in Grid. And a much reasonable trust model named G-Trust model is introduced for Grid. This model is based on the traditional behavior-trust model, and is improved by changing the components of Reputation in the traditional one.

2 Security Hierarchy in Grid

In computational Grids, all users in wide-area network can use Grid resources through logging on Grid. While the fact is that users, resources and applications will not be reliable and beneficial for each other and so Grid security becomes much complicated. In the next parts the analysis about Grid security will be discussed.

2.1 Users Security

Grid users compose a virtual organization (VO). In Grid, many things can be considered as users, such as persons, machines, services, etc. For a legal user, what he cares are two things, 1) whether the resources he requests are usable, 2) if his access right is secure and not seized. These things can be secured by constituting the rules of users management and bi-directional authentications.

2.2 Resources Security

The static authentication mechanism is usually used to judge which rights should be assigned for users. For supporting this mechanism, many techniques are used, as encryption, data hiding, digital signatures, authentication protocol, etc. The advantage of this static control mechanism is simple and feasible and the techniques in it are developed maturely. While Grid is distributed and flexible, this kind of static authentication mechanism contains some faults: 1) during parallel applications running, the dynamically secure mechanism can't be supported. 2) this kind of static mechanism can't be extended feasibly.

For resources owner, the most important thing is resources security. For supporting it, users information are stored and accordingly the resources are assigned. Users information includes Users ID, password, E-mail address, login times, etc. This kind of information about users and resources are usually managed by Grid catalogue, in experimental Grid named Wader, the hierarchical access control for Grid catalogue is implemented by using GBLP [5] secure model, and the hierarchical management for users and resources is supported by this secure access control.

2.3 Applications Security

Applications is the tool which link users and resources. Applications are submitted by users, and apply Grid resources according their functions, then get the results finally. After jobs submissions, in the one hand, for assuring resources security, the submissions must be secure and harmless. For supporting it, a usual way is to insert certain code into jobs for the real-time alternation. In the other hand, for satisfying the demands of users, the application result must be kept secret and not easy to be acquired by other users. These could be achieved by using encryption techniques.

2.4 Network Security

In Grid, supporting secure and feasible network is very important. After a long time study, many mature security techniques are applied to support network security. Such as, *Invading the Detection System*, encryption and decryption of important data, secure channel, etc. How to combine the present security techniques with Grid environment is the key to implement secure and reliable network in Grid. GSI of Globus is the successful symbol of applying the mature techniques to Grid.

2.5 Trust Relationship in Hierarchical Security

From the above analysis, it is feasible to see that the essential action in Grids is that users and resources are linked by applications. So how to establish a reliable trust relationship among users, applications and resources, and how to constitute corresponding secure mechanism and access control according to this relationship, are the base to support Grid security infrastructure. And the concept figure of trust relationship among users, resources and applications is shown in Fig. 1.

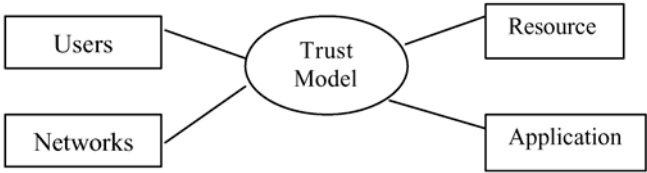


Fig. 1. The concept figure of trust relationship in Grid security

From figure 1, we discover the relationship among users, resources, networks and application is complicated. How to collaborate them into a whole is a challenge problem. Some of these problems separately happen on users, applications and resources, and some of them happen in their mixing processes. So, we need study the trust relationship among them. The trust relation is constructed into a trust model, which is the base to design security infrastructure of Grid. Trust model is the effective way to implement the relationship among users, resources and applications. In the next Section, a trust model named G-Trust Model, which supports security management in Grid, will be described using mathematic definition.

3 Trust Model Based on Behaviors

Trust is the important aspect of study on secure problem. And it is usually divided into two kinds of study: *Identity Trust* and *Behavior Trust*. *Identity Trust* adopts static control mechanism, which limits users access rights before they access and statically stores these rights in a Grid server. *Behavior Trust* judges users trust levels by their history and present behavior, then according to these, users access rights are confirmed dynamically.

Now for solving secure problems of Grid, a trust model named *G-Trust Model* will be introduced in this Section. Before the introduction of this model, some related concepts should be defined firstly.

3.1 The Conception Definitions in G-Trust Model

Trust Level [1]: In special system, special time, special context, Object *A* give single trust score about Object *B* from the present direct touch with it, Remarkd as: $DTT(X, Y, c)$, where *X* and *Y* represent the two Objects which have direct touch, and *c* is the context.

Direct Trust [1]: In special system, special time, special context, Object *A* give the trust score about Object *B* from the history behavior of direct touch with Object *B*. Remarkd as : $\Theta(X, Y, t, c)$, where *X* and *Y* represent the two Objects have the direct touch, and *c* is the context.

Reputation: The indirect trust based on recommending, expresses in special system, special time, a set of objects haven't direct touch with the given object at present time but have some these touches before, give the direct trust to the given object. Similarly, it is remarkd as $\Omega(X, Y, t, c)$, with the same meaning of X, Y, t, c .

Direct Trust Score: In special system, special time, special context, Object *A* give a trust score to Object *B* according to all of their historic direct touches. Remarkd as: $\Delta(X, Y, t, c)$, with the same meaning of X, Y, t, c .

Attenuation Function: in special system, special time, special context, from the last direct touch or updating, the physical reduced level of direct trust score or reputation. Remarkd as: $T(t - t_{ir}, c)$, where *t* is the present time, and t_{ir} is the time of last updating or direct touch, *c* is the related context.

Attenuation Function of Trust Relationship: The physical reduced level of trust relationship about any two entities in system like *X, Y*. Remarkd as: $D(t - t_{ir}, c)$, where t, t_{ir}, c have the same meanings as the above.

Acceptable Image [1]: After overall trust, the final conclusion drawn by system is concerned with if assign resources to requester.

3.2 The Description of G-Trust Model

Function 1. In a special system L , for $\forall$ Object X , $X \in L$, $\exists$ Object Y , at special time t , if X and Y have direct touch, then

$$\Theta(X, Y, t, c) = DTT(X, Y, c) \cdot T(t - t_{ir}, c) \quad (1)$$

Function 2. In system L , for $\forall$ Object X , $X \in L$, $\exists$ Object Y , at special time t , if $\exists t_i \prec t$, with $i \in (1, 2, \dots, n)$, and $DTT(E_i, E_r, c)$ exists, then

$$\begin{cases} \Delta(X, Y, t, c) = \sum_{i=1}^n \alpha_i \Theta(X, Y, t_i, c) \\ 0 < \alpha_i < 1 \\ \alpha_1 + \alpha_2 + \dots + \alpha_n = 1 \end{cases} \quad (2)$$

Function 3. In any system L , for $\forall$ Object X , $X \in L$, $\exists$ Object Z , then $\exists S = \{Y_i | Y_i \in L, \text{ and } \Theta(E_Y, E_Z, t, c) \text{ exists}\}$, with $i \in n$ and $X \notin L$, then

$$\begin{cases} \Omega(E_X, E_Z, t, c) = \sum_{i=1}^n \beta_i (RE(X, Y_i), m) \Delta(E_{Y_i}, E_Z, t, c) \\ 0 < \beta_i < 1 \\ \beta_1 + \beta_2 + \dots + \beta_n = 1 \\ m \in N \end{cases} \quad (3)$$

Function 4. In any system L , for $\forall$ Object X , $X \in L$, then

$$\begin{cases} \Gamma(E_X, E_Z, t, c) = \gamma_1 \Delta(E_X, E_Z, t, c) + \gamma_2 \Omega(E_X, E_Z, t, c) \\ 0 < \gamma_1, \gamma_2 < 1 \\ \gamma_1 + \gamma_2 = 1 \end{cases} \quad (4)$$

3.3 The Analysis of G-Trust Model

G-Trust model is introduced by improving behavior-trust model at the following aspects.

(1) The concept of direct trust score is introduced. The direct trust in primary model assignments trust level is only according to the last direct touch, and it is not enough to judge if a user is believable or not. For example, a user has an untrust history behaviors, while if he behaves legally in once single login, then in the next time, he will be assigned a higher trust level to apply important resources, and it's dangerous for those resources which would not be assigned to him. While by inducing the concept of direct trust score the history behaviors of users are considered. In another words, the users behaviors are tracked available, then the possibility of wrong resources assignment is reduced. And accordingly the reliability of overall trust is improved.

(2) The concept of Attenuation Function of trust relationship is introduced for physically showing with time the change of trust relationship among resources nodes. Since Grids are dynamic, this concept exactly reflects this kind of dynamic trust relationship among resources.

New components of reputation are established. In the primary model, reputation is stably shows as reputation table. This reputation may be effected directly by the subjective factors of reputation supporter. And the Grid environment is dynamic, this stable reputation can't exactly express the present relationship among resources nodes.

(3) In this model, many parameters are used, such as $\alpha_i, \beta_i, \gamma_i$, they express the different influence respectively about the behavior trust attenuation by time, change of relationship among resources nodes and the different statuses between direct trust score and reputation in evaluating the overall trust. For the universality, we don't introduce a certain rule to restrict the change of these parameters in the mathematical description. According to different characters of Grid system, these parameters can be restricted with different rules. In the follow use case, the parameters are defined simply and the whole construction is used.

4 The Grid Security Framework with Trust Model

The relationship of Users, applications and resources is managed using trust model. And by constructing the grid security framework, the initialization, modification, and updating of trust are all achieved. In this framework, there are two modules: inspection module is responsible for investigating trust levels, evaluation module is responsible for evaluating trust levels. Figure 2 shows the structure of a factual grid security framework with the trust model defined above.

4.1 LDAP Server

Grid information server, which is an important component of Grid infrastructure, is used to manage meta-data. This saved data in it mainly includes the information about users, resources nodes, network and computing. Because of the feasibility and security, LDAP[6] is usually used as information server in large-scale parallel system, e.g. Globus, Netsolve[7], etc.

In the security infrastructure of our experimental Grid (*WADER*), LDAP is used as Grid information Server, and at the same time, it also plays a security role which is responsible for saving resources trust evaluation for users as their secure levels and mapping these secure level to their access rights by ACL (*access control list*). When clients send resources request to Grid portal, the Grid information server will assign corresponding resources to them according to users access rights and the present state of resources. Once the resources assignment succeeds, broker will broadcast the messages to resources domain according to the factual assignment scenario.

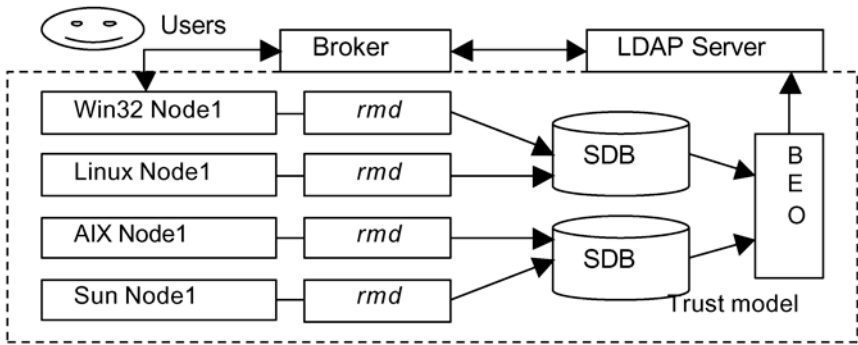


Fig. 2. The grid security framework with trust model

4.2 Evaluation Module

Evaluation module includes two parts: SDB (*security database*) and BEO (*behaviors evaluation organization*). SDB is responsible for saving the users present and history behaviors. There are many SDB separately distribute in different resources domains. Here, Mysql server is used as SDB because it supports heterogeneous systems and is easy to operate. BEO is the core component of this security infrastructure. According to users present and history behaviors saved in SDB, BEO calculates users trust level by the above G-Trust model. And once the new trust level is calculated, BEO will submit it to Grid information server for updating the history.

Evaluation module is responsible for inspecting the using instance of resources and collecting the resources trust evaluations for users. The function of inspecting resources and collecting trust evaluations is realized by resided the process *rmd* on each node. *rmd* is saved on nodes, when broker assigns jobs to resources, *rmds* on used resources are running and then send the results of trust evaluations to SDB.

5 Conclusion

Grids are applied as new infrastructure, which can support parallel computing in distributed computational resources, and an indispensable study part in it is Grid security. How to secure resources safely and validly is the hotspot in the study field of Grid security. Trust model is a secure model based on behavior-trust, and it assigns resources by users history behavior. In this paper, *G-Trust Model* is introduced from

traditional trust model. And by applying new components of reputation, improved *G-Trust Model* is more comparable to the security requirement of Grid. Though resources in any system are all not as the same, and different resources have the different demand for security. How to insert dividing secure levels for resources into this security infrastructure is the next step of Grid security.

References

1. Azzedin F, Maheswaran M. Evolving and managing trust in Grid computing systems[C].In: Canadian Conference on Electrical and Computer Engineering, IEEE CCECE 2002, Volume 3, 2002, pp.1424-1429
2. A Abdul-Ranman, S Hailes Supporting trust in virtual communities[C]. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. vol.1, 2000, pp.1-9
3. <http://www-unix.globus.org/security/>
4. <http://www.cs.virginia.edu/~legion/>
5. Bing Xie, Xiao-Lin Gui, Qing-Jiang Wang Study on a Secure Access Model for the Grid Catalogue, LNCS, Vol.3032, 2004, pp.891~898
6. H. Johner, L. Brown, F. Hinner et al. Understanding LDAP, International Technical Support Organization, Jun1998
7. N. Casanova, H. Dongarra. Enabled Solvers for Scientific computing Using the NetSolve System, 3rd International Conference on Algorithms and Arichitectures for Parrallel Processing, 1997, pp. 17-33

Research on a Quantitative Security Risk Assessment Approach in Large-Scale Early Warning System*

Lei Xuan and Xin Xu

School of Computer, National University of Defence Technology
410073 Changsha, Hunan, P.R. China
xllily1@sina.com

Abstract. Large-scale Early Warning (EW) is an indispensable component for protecting national information infrastructure. Various qualitative and quantitative models of Security Risk Assessment (SRA) are surveyed and evaluated in this paper. Then, the paper proposes a hierarchical on-line SRA model for three levels of subsystems in an EW system, i.e., local EW groups, regional EW centers, and the national EW center. In this model, the SRA system in a regional EW center evaluates the threat, vulnerability, impact, and control of each local group to calculate the local residue risk value, and calculates the regional residue risk value and reports it to the national EW center. To compute the national residue risk value, the SRA system in the national EW center synthesizes reports and values from all regional centers. A prototype of the hierarchical on-line SRA model was implemented in an EW system. Experimental results show the effectiveness of the proposed method.

1 Introduction

With the increasing threats of network attacks and Information Warfare (IW), it becomes indispensable for protecting national information infrastructure to develop and establish an effective and reliable information security Early Warning (EW) network. Because the infrastructure is a large-scale system and its security is uncertain and dynamic, the EW network needs to be built both from the top down with a national EW center and from the bottom up with local/regional EW groups or centers. By integrating both local/regional and national capabilities, a robust and rapid EW system with the capacity for a wide range of threats can be created.

In addition, it is significant to distribute EW in different levels, because response system must make decisions according to the level of EW. In order to assess the level of EW, the Security Risk Assessment (SRA) of information systems is adopted in our EW system. The security trends and potential threats can be assessed on the collection and analysis of data obtained through open-source collection. Open-source materials can be obtained from the Internet and news sources, which can provide valuable information for planning, training, and preparation efforts for managing the consequences of infrastructural attacks^[1].

* Work is supported by the National Natural Science Foundation of China, under Grant 60303012, 90104001 and National High Technology (863) Program No. 2003AA142010.

In this paper, we analyze some typical models in security standards and framework [2]~[7] at first. Then the methods [8]~[18] of qualitative and quantitative SRA of information systems are overviewed. Based on the overview, it can be concluded that existing models and methods mainly focus on security investment decision-making, so they are not dynamic but static. The models can only be used in security evaluation for one organization, and not suitable for large-scale networks. EW system differs from security investment decision-making in that: 1) EW needs level assessment in time, 2) EW is large-scale, which includes many local groups and regions, 3) EW must make decisions based on preliminary assessment about the effect of existing security controls, and 4) EW must consider the international relationship threat elements responsible for IW. Having considered all these characteristics, this paper presents a hierarchical model of on-line SRA for three classes of EW systems: local EW groups, regional EW centers, and the national EW center. This paper concentrates mainly on the model design and algorithms of the SRA in EW system.

2 Related Work

Security risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization. SRA is the process of identifying the risks to system security and determining the probability of occurrence, the resulting impact, and additional safeguards that would mitigate this impact. Risk management is the overall process of identifying, controlling and mitigating information-system-related risks. [8] It includes security risk assessment, cost-benefit analysis, and security defense policy. The security defense policy contains selection, implementation, test, and evaluation. The security management in information systems can be viewed as the process of taking steps to reduce risk to an acceptable level. CC [2], SSE-CMM [3], ISO/IEC 17799 [4] (BS 7799[5]), ISO/IEC TR 13335 [6], and IATF [7] all treat risk management as a main part in security management.

2.1 Risk-Management Models of Information Security

ISO/IEC TR 13335 [6] presents its own model of the relationships between security elements which are often associated with risk management. The main elements involved in security management contain assets, threats, vulnerabilities, impact, risk, safeguards, residual risk, and constraints. Similarly, CC [2] defines the high level concepts and relationships of security that may be involved in risk management. CC emphasizes the countermeasure evaluation, and the outcome of evaluation is a statement about the extent to which assurance is gained that the countermeasures can be trusted to reduce the risks to the protected assets. This statement can be used by the owner of the assets in deciding whether to accept the risk of exposing the assets to the threats.

SSE-CMM model [3] defines four Process Areas (PAs) about security risk clearly: PA04—assess threat, PA05—assess vulnerability, PA02—assess impact, and then PA03—assess security risk. Comparatively, IATF [7] goes further than other models. It proposes that risk management should be applied during the initial system devel-

opment, throughout the development, acquisition process. The risk management process needs to be adjustable and respond to those elements that cause a resulting change in risk, i.e., changes in the system design and configuration, and changes in the operating environment. On this account, risk management is cyclical in nature.

Through the analysis on above models, we found that there are three important factors which may lead to security risks: system vulnerability, threat event, and the resulting impact on assets from event. Generally speaking, security risk will be presented only when the three factors exist simultaneously (i.e. each risk factor is over 0).

2.2 Methods of SRA for Information Systems

An information system has security risks due to the vulnerability inside the information system and threats from IW and various attacks. The network mission impact considering (1) the probability that a particular threat-source will exercise the particular information system vulnerability, and (2) the resulting impact if this should occur [8]. The assessment on the both factors at the same time belongs to **quantitative** SRA, while the assessment only on impact belongs to **qualitative** SRA.

Qualitative SRA. The OCTAVESM [9] is a risk-management process that helps security managers to identify their threats and vulnerabilities containing three phases. The NIST [8] recommends that government agency use their qualitative risk-management process. Although the two SRA process eventually results in a quantitative evaluation of risks, the management method is more qualitative than quantitative. Security managers use three levels of assessment—high, medium, and low—to establish the likelihood and impact of a threat-vulnerability realization.

COBRA [10] serves SRA mainly in terms of system vulnerability, threat, impact, and security control measure. In the case of qualitative assessment, their relationship is represented by attacks. Threat (T) creates the attack against the information system, which exploits the system vulnerability (V). The existence of vulnerability results in impact (I) if attacks happened. Different controls (C) have respective functions to mitigate risks. The result on security Total Risk (TR) value caused by these elements can be represented qualitatively as:

$$TR = T \times V \times I . \quad (2.1)$$

At the same time, the Residual Risk (RR) is:

$$RR = T \times V \times I \div C . \quad (2.2)$$

As a result, the main challenge in qualitative SRA is to quantify these four risk elements. Fuzzy mathematics and artificial intelligence are introduced into SRA domain because of the subjectivity and uncertainty in assessing. The ICISA of King's College London established an intelligent threat assessment model in the IWAAS [11]. This model adopts technologies of expert systems and intelligence fusion to assess IW threat. However, vulnerability assessment has experienced the stage of manual-to-automatic, now expanding from partial assessment to holistic, from rule-based to model-based, from single-host to distributed [12].

Quantitative SRA. Quantitative SRA makes use of a single figure produced from two elements: the probability of an event occurring and the likely loss should it occur. This is called the 'Annual Loss Expectancy (*ALE*)' or the 'Estimated Annual Cost (*EAC*)'. In *ALE*, the cost-benefit of a risk mitigation control equals to the difference between the *ALE* with and without the control, minus the cost of the control ^[13]. *ISC*² (means I S C squared) ^[14] and reference [15] recommend that security practitioners use a quantitative risk-management method based on *ALE*.

Before computing the *ALE*, it must determine the Asset Value (*AV*), the Exposure Factor (*EF*), the Single Loss Expectancy (*SLE*), and the Annualized Rate of Occurrence (*ARO*). The *EF* is the percentage loss that a realized threat event would have on an asset. The *ARO* is an estimation of the probability that a threat will occur during a year. Then it computes the *ALE* using the *SLE* and *ARO*:

$$SLE = AV \times EF . \quad (2.3)$$

$$ALE = SLE \times ARO . \quad (2.4)$$

After computing the *ALE* of threat-vulnerability-asset combination, (*ISC*)² recommends to conduct a cost/benefit analysis to determine the value of a risk-mitigation control as follows:

$$(Value\ of\ control) = (ALE\ pre-control) - (ALE\ post-control) - (Annual\ cost\ of\ control). \quad (2.5)$$

Managers can make security investment decision-making according to the *value of control*.

It has developed some software packages for automated information risk analysis, assessment and management from 1980s, like @RISK, BDSS (Bayesian Decision Support System), CRAMM (Critical Risk Analysis and Management Method), and RiskCALC ^[16,17]. They are all hybrid methods ^[18], i.e., some selected combinations of qualitative and quantitative methods can be used to implement the components utilizing available information while minimizing the metrics to be collected and calculated.

The qualitative assessment is simpler and widely used. It uses simple calculations and uses procedure in which it is not necessary to determine the dollar value of all assets and the threat frequencies or the implementation costs of the controls. Quantitative assessment does this as well as identifies the specific envelope in which the losses and safeguards exist. It presents its results in a management-friendly form of monetary values, percentages, and probabilities. The hybrid model uses a facilitated risk analysis process which is gaining in popularity due to its reduced costs and efforts required.

3 The Design of a Hierarchical On-Line SRA Model

It is significant to give out an EW in time, because the information system must respond rapidly according to the level of EW. In order to assess the level of EW, the SRA method of information systems is adopted in our EW system.

EW system differs from security investment decision-making in four characteristics as introduced in 1. Having considered all requirements of EW systems, a quantitative method is necessary in the system, and this paper presents a hierarchical model of on-line SRA in three levels of EW sub-systems: local EW groups, regional EW centers, and the national EW center, which is illustrated in figure 1.

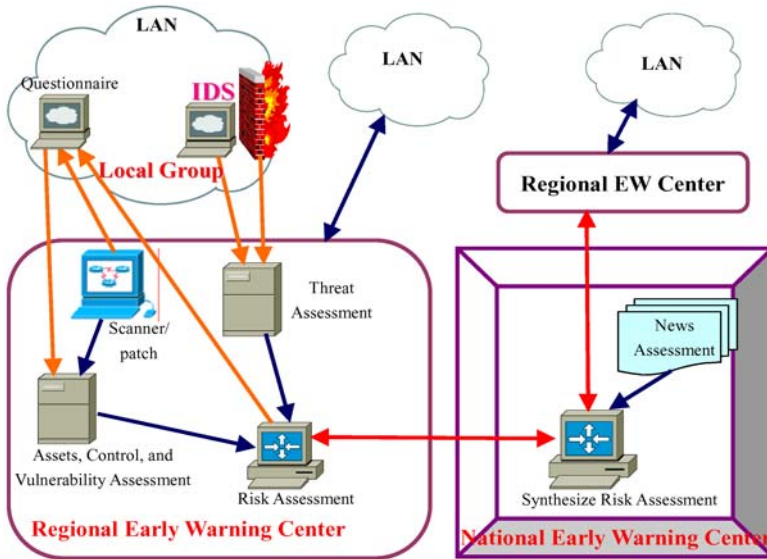


Fig. 1. The hierarchical model of on-line SRA in Early Warning system

Data resources come from IDS, firewall, scan systems, user questionnaire, and international news data, while processes contain threat assessment, assets/impact assessment, control assessment, vulnerability assessment, news assessment, and risk assessment. The system outputs are the risk value, risk assessment report, warning and suggestion.

The main function of threat assessment is the threat probability of network attacks occurrence. The attacks are detected by IDS and/or firewall of every registered local group and the statistics of threat probability are calculated separately for every local group. The threat probability of attack i in local group j is $P_{i,j}$.

The function of vulnerability assessment is figured out by firstly scanning the local network in search of vulnerabilities, and then quantifying the possible Exposure Factor of impact on one asset by the vulnerability for each threat-vulnerability-asset triplet i , as $EF_{i,j}$.

However, the assets assessment and control assessment nowadays still need user questionnaires for data acquisition. The assessment value can be reached only after statistical weights. The value of asset i in local group j is $AV_{i,j}$, and the Control Gap (Risk left under current control policy.) is $CG_{i,j}$.

Now the SRA of the regional EW center can calculate the residue risk of threat-vulnerability-asset triplet i in local group j , as $RR_{i,j}$, which can be represented as :

$$RR_{i,j} = AV_{i,j} \times EF_{i,j} \times P_{i,j} \times CG_{i,j}. \quad (3.1)$$

Then it makes descending sort and chooses the top N (twenty or so, decided by user's experience) by $RR_{i,j}$ from local group j . Since threat-vulnerability-asset triplets are associated each other, so the residue risk of local j , DRR_j , can be represented as formulation of risk factors being in dependent case:

$$DDR_j = \sqrt{\frac{\sum_{s=1}^N RR_{s,j}^2}{N}}. \quad (3.2)$$

And then it will calculate the residue risk of region (RRR). It still uses the formulation of dependent case for security risks being associated with each other. If it has m local groups registered in regional center k , the RRR_k is:

$$RRR_k = \sqrt{\frac{\sum_{j=1}^m DDR_j^2}{m}}. \quad (3.3)$$

The regional EW center will alarm according to the RRR_k value, and generate assessment reports to the national EW center, and give risk control suggestions to every local they protected.

The threat elements of international relationships responsible for IW belong to the strategic field and only need to be processed in national EW center. The main function of news assessment is the calculation of threat weight value of attack caused by the international relationship threat elements. But international relationship threat elements are related to the fields like technology, strategies, economics, and politics etc. So, it is necessary to integrate information from multiple resources like the above-mentioned fields in the value. And since the information is characterized as uncertain, incomplete, fuzzy, and dynamic, the quantification of this information is provided in a subjective manner. In [19], the authors transformed the political, economic, cultural, and strategic information from various resources to float numbers between [0,1] by adopting a fuzzy and empirical method. Then it designs a *Mamdani* fuzzy neural network reasoning algorithm to speculate the acquired information from diverse fields, and computes the weight value of international relationship threat element, as TW .

The SRA of national EW center synthesizes every regional center residue risk RRR_k and the weight value TW to assess the security risk of the whole information system. The national center still uses the formulation of dependent cases for the same reason, if it has r regional centers, the residue risk of nation (NRR) is:

$$NRR = \sqrt{\frac{\sum_{k=1}^r RRR_k^2 + TW^2}{r+1}}. \quad (3.4)$$

The national EW center will alarm every regional EW center according to the NRR value, and generate assessment report and risk control suggestions. The ultimate purpose is to, by means of combining the power of national center and respective regional centers, set up a dependable and efficient EW network that can be responsive to threats over a wide range of fields.

A prototype of the SRA was implemented in an experimental EW system. In the experiments, the regional EW center can give the $RR_{i,j}$ in 1 second, and calculates the DDR_j , RRR_k and generates report in every minute. The national EW center also com-

puts *NRR* and generates report in every minute. So the proposed SRA method can assess the risk of the network timely to support EW decision-making efficiently.

4 Conclusion and Future Work

In order to assess the levels of EW in time, based on the research on various models and methods in SRA, this paper proposes a hierarchical on-line SRA system model for three levels of EW sub-systems, i.e., local EW groups, regional EW centers, and the national EW center. A prototype of the hierarchical on-line SRA model was implemented in an EW system. Experimental results in the prototype show that the method can assess the levels of EW in time to support EW decision-making efficiently. The future research will focus on the corresponding relationships of “threat-vulnerability-assets” triplet and the joint probability of threat-vulnerability to realize more creditable risk value so that the EW will become more pertinent and the decision-making will be more concrete.

References

1. Sandia National Laboratories: Us Infrastructure Assurance Strategic Roadmaps – Strategies For Preserving Our National Security. August (1998)
2. The International Organization for Standardization: Common Criteria for Information Technology Security Evaluation. ISO/IEC15408: 1999(E) (1999)
3. SSE-CMM Model Description Document Version 2.0. <http://www.sse-cmm.org> (1999)
4. International Organization for Standardization: BS ISO/IEC 17799:2000 Information Technology – Code of Practice for Information Security Management. December (2000)
5. BSI/DISC Committee BDD/2: BS7799 Code of Practice for Information Security Management (1999)
6. International Organization for Standardization: ISO/IEC TR 13335, Information Technology – Guidelines for the Management of IT Security (GMITS) (1996-2001)
7. National Security Agency: Information Assurance Technical Framework (IATF), Version 3.0. <http://www.iatf.net> (2000)
8. Gary Stoneburner, Alice Goguen, Alexis Feringa: Risk Management Guide for Information Technology Systems. Recommendations of the National Institute of Standards and Technology (NIST). NIST Special Publication 800-30. October (2001)
9. Alberts, Christopher J., Dorofee, Audrey J.: OCTAVESM Method Implementation Guide, v2.0. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University (2001)
10. C&A Systems Security Ltd.: COBRA: Introduction to Risk Analysis. <http://www.ca-systems.zetnet.co.uk/risk.htm>
11. Andrew Rathmell, Richard Overill, Lorenzo Valeri: Information Warfare Attack Assessment System (IWAAS). Information Warfare Seminar. London. October (1997)
12. Xing Xujia, Lin Chuang, Jiang Yixin: A Survey of Computer Vulnerability Assessment. Chinese Journal of Computers, Vol.27, No.1. Jan (2004) 1-11
13. Shawn A. Butler: Security Attribute Evaluation Method. CMU-CS-03-132. Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy. School of Computer Science Carnegie Mellon University, Pittsburgh. PA 15213. May (2003)
14. International Information System Security Certification Consortium (ISC)²: Security management. <http://www.os-global.com/>

15. Ding Tan.: Quantitative Risk Analysis Step-By-Step. GSEC Practical Version 1.4b – Option 1. As part of the Information Security Reading Room. SANS Institute (2003)
16. Thomas R. Peltier.: Information Security Risk Analysis. New York: Auerbach Publications (2001)
17. List of Risk Analysis, Assessment and Management Tools.
<http://www.theiia.org/ITAudit> (1998)
18. James W. Meritt, CISSP.: A Method for Quantitative Risk Analysis.
<http://www.auditnet.org/docs/riskmeth.PDF>
19. Miao Qing.: Characteristic Information Fusion Method on Network Security Strategic Indication/Warning System. Computer Engineering, Vol.28, No.7 (2002)

A Formal Logic for Shared Resource Access Control in the Grid*

Baiyan Li, Ruonan Rao, Jinyuan You, and Minglu Li

Department of Computer Science and Engineering, Shanghai Jiao Tong University
Shanghai 200030, China
libaiyan@sjtu.edu.cn

Abstract. This paper presents a formal logic that can be used to model security mechanisms associated with the access-control of shared resources in the grid environment. The logic uses the $\mathbf{K45}_n$, a standard modal logic of belief, and a fine-grained *trust* relationship to describe and reason about the access-control related issues. In this paper, the motivation, syntax, semantics, inference rules of the logic as well as how to encode credentials and security policies using the logic are introduced. An example that demonstrates how to use the logic in authorization decision making for resource requests within grid environment is also given.

1 Introduction

The primary purpose of the grid computing is to share resources dynamically within Virtual Organizations (VOs) [1, 2]. So the resource protection is clearly a critical task for a secure grid environment. However, a grid is an open, large-scale, distributed system. The challenges, the access-control system faced in a grid environment, are quite different from that in the traditional centralized, or relatively small distributed systems that are based on the closed-world assumption [3]. In contrast to traditional systems, the grid environment has following inherent properties: (1) The access-control mechanisms of shared resources are decentralized. (2) A grid is a distributed system across multiple administrative domains. The grid-based applications need global security services rather than small or organizational ones. (3) Because there is no a central authority that everyone trusts in a grid environment, resource owners must use the information from third parties they trust to make decision for resource requests from some strangers.

The inherent properties of the grid arouse some security related problems that have been put forward in researches of *PKI interoperability* [4, 5] and *trust management* [3, 6–8]. For traditional centralized systems, there exist several access-control models, such as the Bell-LaPadula model, in the literature. Similarly, the counterpart is also needed for the grid system. In order to characterize the

* This paper is supported by 973 project (No.2002CB312002) of China, and grand project of the Science and Technology Commission of Shanghai Municipality (No. 03dz15027 and No. 03dz15028).

access-control system for the grid, we propose a formal logic based on many well-known contributions [9–14] in the literature. The logic can be used to encode various credentials and security policies in logical formulas. Its inference rules can be used to build a compliance checker [8] that accepts the logical formulas as input and makes authorization decision for resource requests. This paper introduces the theoretic basis of the logic, and demonstrates the encoding method of credentials as well as authorization decision making using the logic.

The rest of this paper is organized as follows. In Section 2, the syntax, semantics, and inference rules of the logic are introduced. In Section 3, the encoding method of credentials is given. In Section 4, an example for using the logic is presented. In Section 5, some related works are discussed. Finally, we give the conclusion of this paper and future works in Section 6.

2 The Logic

There is no a widely accepted definition of *trust* in the literature [15]. We now define a special trust relation in terms of *belief*, and develop a formal logic of trust based on modal logic $\mathbf{K45}_n$ and *possible-worlds* semantics [14] in the section.

2.1 Syntax and Semantics

A logic of any kind needs a language to define its well formed formulas (*wffs.*). Given a nonempty set Φ of primitive propositions, which we typically label $p, q, \dots$, and a set of n agents whose names are denoted $A_1, A_2, \dots, A_n$, we define a modal language $\mathcal{L}_n(\Phi)$ to be the least set of formulas containing Φ , closed under negation, conjunction, and modal operators $B_1, B_2, \dots, B_n$. In other words, the formulas of $\mathcal{L}_n(\Phi)$ are given by rule $\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid B_i p$ where p ranges over elements of Φ and $i = 1, \dots, n$. We use the classical abbreviations $\varphi \vee \psi$ for $\neg(\neg\varphi \wedge \neg\psi)$ and $\varphi \supset \psi$ for $\neg\varphi \vee \psi$; we take *true* to be an abbreviation for some valid formula such as $\neg p \vee p$, and define *false* as the negation of *true*. Especially, the modal operator B_i in the rule is read as “agent A_i believes”. In this paper, the formulas $B_i p$ and A_i *believes* p are semantic equivalence, and can be used alternatively for the sake of convenience. Beside the modal operator *believes*, operator *says* is often used to represent an agent actually making a statement in the literature [12]. So the *says* modality is performative. For a statement ϕ , if A *says* ϕ then A *believes* ϕ . However, it is not the case conversely. In order to interpret the semantics of formulae in $\mathcal{L}_n(\Phi)$, we define a semantic model for the language.

Definition 1. A frame for modal language $\mathcal{L}_n(\Phi)$ is a tuple $\Gamma = (S, \mathcal{K}_1, \dots, \mathcal{K}_n)$ where S is a nonempty set, and $\mathcal{K}_i$ is an accessibility relation set on S for $i = 1, \dots, n$. A model for $\mathcal{L}_n(\Phi)$ is a pair $M = (\Gamma, \pi)$, where Γ is a frame, and π is a truth assignment to the primitive propositions in Φ for each states $s \in S$ (i.e., $\pi(s) : \Phi \rightarrow \{\text{true}, \text{false}\}$ for each state $s \in S$).

The model M , usually denoted $M = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$, is a typical Kripke structure for n agents [9]. Intuitively, we say that $(s, t) \in \mathcal{K}_i$ iff agent A_i considers

state t possible at state s where $s \in S$ and $t \in S$. A formula in $\mathcal{L}_n(\Phi)$ is true at a state s in a model $M = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$ can be defined as following:

$$\begin{aligned} (M, s) &\models p && \text{iff } \pi(s)(p) = \text{true} \text{ (for } p \in \Phi) \\ (M, s) &\models \varphi \wedge \psi && \text{iff both } (M, s) \models \varphi \text{ and } (M, s) \models \psi \\ (M, s) &\models \neg\varphi && \text{iff } (M, s) \not\models \varphi \\ (M, s) &\models B_i\varphi && \text{iff } (M, t) \models \varphi \text{ for all } t \text{ such that } (s, t) \in \mathcal{K}_i \end{aligned}$$

The first three definitions correspond to the standard clauses in the definition of truth for propositional logic. The last definition formalizes the idea that agent A_i believes φ in global state s exactly if and only if φ is true in all the global states that A_i consider possible in state s . Formally, we say that a formula φ is valid in M , and write $M \models \varphi$, if $(M, s) \models \varphi$ for every state $s \in S$; we say that φ is satisfiable in M if $(M, s) \models \varphi$ for some state $s \in S$. We say φ is valid with respect to a class $\mathcal{M}$ of structures and write $\mathcal{M} \models \varphi$, if φ is valid in all structures in $\mathcal{M}$, and say φ is satisfiable with respect to $\mathcal{M}$ if it is satisfiable in some structure in $\mathcal{M}$. We use $\mathcal{M}_n$ to denote the class of all Kripke structures for n agents.

In this paper, we adopt the well-known **K45_n** as axiom system for our logic. The soundness and completeness of **K45_n** with respect to $\mathcal{M}^{et}$, the class of transitive and Euclidean Kripke structures, has been well-known [14]. So it provides us a substantial basis for modelling belief and trust. We now give definition of a strong trust relationship between agents in a $\mathcal{M}^{et}$ structure.

Definition 2. Let $M^{et} = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$ be a transitive and Euclidean Kripke structure. Let A_i and A_j ($1 \leq i, j \leq n$) be a pair of agents in $\mathcal{M}^{et}$. Let R be a formula set of $\mathcal{L}_n(\Phi)$. We say that A_i strongly trusts A_j regarding R , denoted by formula $A_i \succ_R A_j$, if and only if $(B_j\varphi \supset B_i\varphi) \wedge (B_j\neg\varphi \supset B_i\neg\varphi)$ where variable φ ranges over elements of R .

In this way, we defined a restricted trust relation between agents in term of the belief. Before giving the semantics of the trust relation with respect to model M^{et} , we define a set of states for a given state $s \in S$

$$\mathcal{K}_i(s) = \{t \mid \forall t \in S \text{ if } (s, t) \in \mathcal{K}_i\}, \quad i = 1, \dots, n.$$

We also define a mapping $\theta^* : S \rightarrow S$ that takes states to a subset of the original states where “*” represents some set of formulae. Given a set $S_0 \subseteq S$ and a set R of formulae, we define

$$\theta^R(S_0) = \{s_0 \mid \forall s_0 \in S_0, \text{ if } (M^{et}, s_0) \models \varphi \text{ for all } \varphi \in R\}.$$

We now use notation $\neg R$ to represent the negation of R . So we have that $\neg R = \{\neg\varphi \mid \forall \varphi \in R\}$. Then, the semantic definition of strong trust-regarding relationship is given as following:

$$\begin{aligned} M^{et} \models A_i \succ_R A_j &\text{ iff } \forall s \in S, \theta^R(\mathcal{K}_i(s)) \subseteq \theta^R(\mathcal{K}_j(s)) \text{ and} \\ &\theta^{\neg R}(\mathcal{K}_i(s)) \subseteq \theta^{\neg R}(\mathcal{K}_j(s)). \end{aligned}$$

By the relationship between modal operator “believes” and “says”, we define a weak trust relation between agents as following:

Definition 3. Let $M^{et} = (S, \pi, \mathcal{K}_1, \dots, \mathcal{K}_n)$ be a transitive and Euclidean Kripke structure. Let A_i and A_j ($1 \leq i, j \leq n$) be a pair of agents in M^{et} . Let R be a formula set of $\mathcal{L}_n(\Phi)$. We say that A_i weakly trusts A_j regarding R , denoted by formula $A_i \succeq_R A_j$, if and only if $((A_j \text{ says } \varphi) \supset B_i \varphi) \wedge ((A_j \text{ says } \neg \varphi) \supset B_i \neg \varphi)$ where variable φ ranges over elements of R .

In fact, the *says* modality represents an agent actually making a statement, and is never automatically inherited by other agents. On the contrary, the *believes* modality is inherited transitively. So the *strong* trust relationship is transitive while the *weak* trust relationship is not. The symbols “ $\succ$ ” and “ $\succeq$ ” are the abbreviation of “ $\succ_R$ ” and “ $\succeq_R$ ” if R is the universe of all wffs. of $\mathcal{L}_n(\Phi)$.

2.2 Inference Rules

We now give the major inference rules of the logic of trust as following:

$$\begin{array}{ll}
 IR1 : \frac{B_i(A_i \succ_R A_j) \quad B_i B_j \varphi}{B_i B_i \varphi} & IR2 : \frac{B_i(A_i \succeq_R A_j) \quad B_i(A_j \text{ says } \varphi)}{B_i B_i \varphi} \\
 IR3 : \frac{B_l(A_i \succ_R A_j) \quad B_l(A_j \succ_V A_k)}{B_l(A_i \succ_{R \cap V} A_k)} & IR4 : \frac{B_l(A_i \succ_R A_j) \quad B_l(A_j \succeq_V A_k)}{B_l(A_i \succeq_{R \cap V} A_k)} \\
 & IR5 : \frac{B_i(A_i \succ_R A_j) \quad B_j(A_j \succ_V A_k)}{B_i(A_j \succ_{R \cap V} A_k)}
 \end{array}$$

In the above rules, notations A_i , A_j , and A_k ($1 \leq i, j, k, l \leq n$) represent agents. Letters R and V represent formula set of $\mathcal{L}_n(\Phi)$. The symbol φ represents a variable that ranges over elements of R . Because of spaces limitation, we omit the soundness proofs of these inference rules from the paper.

3 The Encoding

In order to encode credentials and security policies using the logic of trust, we devise a special method for representing keys of public encryption system. For instance, $k_A = (k_A^{+1}, k_A^{-1})$ represents a public-key pair of agent A where k_A^{+1}/k_A^{-1} is the public/private part of k_A . If k_A is a public key of agent A , the fact can immediately be encoded in formula $A \succ k_A$. The basic attributes or privileges of agents such as “read file foo”, form the primitive propositions of our logic. If agent A signs a statement, for example φ , with its private key k_A , we express the signed statement as $[\varphi]_{k_A^{-1}}$, and encode it in formula $k_A \text{ says } \varphi$ or directly $k_A \text{ believes } \varphi$.

In our context, the credentials refer to various signed certificates issued by active entities. All of the widely used standard certificates, including identity certificate [16], authorization certificate [17], cross certificate [5], and proxy certificate [4, 16], can be encoded in formulas using the logic of trust.

- **Identity certificate** (x.509 certificate). The identity certificate is used to bind the name of an entity with its keys. For example, if a CA, say CA_1 , signs a certificate to bind k_A with the identity of agent A using its private key $k_{CA_1}^{-1}$, we express the identity certificate as $[A \succ k_A]_{k_{CA_1}^{-1}}$, and encode it in formula $k_{CA_1} \text{ believes } A \succ k_A$.
- **Authorization certificate**. The authorization certificate defined in SPKI [17] is used to bind permissions with public key of an entity. It is often represented by a 5-tuple structure (I, S, D, T, V) where I is the key that issued the certificate; S is the subject of the certificate; D is delegation flag: a boolean value indicating whether this permission may be further delegated; T is authorization: a set of primitive permissions being granted; and V says when the certificate is valid. According to the value of D , we can encode the 5-tuple in formula $I \succ_T S$ ($D=true$) or $I \succeq_T S$ ($D=false$). The deduction from tuple $(I, J, true, A, V)$ and (J, S, D, A, V) to (I, S, D, A, V) [18] is just an application of rule $IR3$ or rule $IR4$ of the logic of trust.
- **Cross certificate**. The cross certificate is used to establish peer-to-peer trust relationship between CAs in different security domains. To capture this trust relationship, we define term id_key as a formula set that contains all the identity-key bindings like $A \succ k_A$. If a CA wishes to limit the trust to its peers, it may specify limitations in the certificates issued to the peers. We apply restrictions to id_key to reflect the constraints such as *name constraints*, *policy constraints* and *path length constraints* [5], in cross certificates. For example, if a CA is restricted to issue certificates only for the subjects in name domain “xyz.com”, then the subjects in domain “abc.xyz.com” satisfy the constraint while those in “abc.zyx.com” not. Obviously, $id\_key.constraint(X) \subseteq id\_key$ is always valid for any PKI domain X . We usually use the principal CA of a PKI domain to represent the domain. Moreover, we use $id\_key.constraint(X_1, X_2)$ to represent the set of identity-key bindings that satisfy both $id\_key.constraint(X_1)$ and $id\_key.constraint(X_2)$ where X_1, X_2 are names of CAs or PKI domains. So if CA_1 issues a cross certificate to CA_2 , we represent the certificate as $[CA_1 \succ_{id\_key.constraint(CA_2)} CA_2, CA_2 \succ k_{CA_2}]_{k_{CA_1}^{-1}}$.
- **Proxy certificate**. The proxy certificate in GSI [2] plays dual roles: one for identity-key binding and one for privilege rights delegation. We use term $id\_key.pc$ to represent the set of identity-key bindings only for the proxy. Hence a proxy certificate can be represented in the form of $[proxyB \succ_{k_{proxyB}}, proxyA \succ_R proxyB, proxyA \succ_{id\_key.pc} proxyB]_{k_{proxyA}^{-1}}$.

The security policies in distributed system have various forms varying from simple to complicated. For simple policies, we can express them by simple trust relationships between entities. For complicated security policies, we can express them using the mechanism of the Role-based Access Control (RBAC) [19] with hierarchy. The details related to this issue will be introduced in another paper.

We adopt the Lampson’s method [10] to deal with certificate expiration, i.e., the formulas that encode a certificate would only be valid in the lifetime of the certificate. We assume that an agent will never issue a negative credential such

as k_A is not a public key of agent A . This assumption ensures the monotonicity of our reasoning system. If an issue cancels some certificate, we just treat that certificate as a time expired one.

4 Using the Logic

Like other formal logics in [10–12], our logic can be used to verify security properties of access-control systems. However, in this paper, we focuses on the use of the logic as a theoretic basis for constructing a trust engine. The trust engine, namely compliance checker [8], is responsible for evaluating credentials and security policies, and answers questions like this: “Should request r be granted under policy set P and credential set C . We now demonstrate how the trust engine can make authorization decision for a resource request by logical inference.

In our scenario, two independent organizations, A and B , decide to closely cooperate with each other for business reasons. Both the organizations originally had their own PKI domains and CAs: $CA1$ of A and $CA2$ of B . For the sake of collaboration, $CA1$ and $CA2$ issued cross certificates to each other. User *Alice* in organization A has a signed identity certificate issued by her trusted root CA, namely $CA1$. Similarly, user *Bob* in organization B has a signed identity certificate from $CA2$. Now *Alice* controls a set of resources, namely S , and *Bob* is a valid user in *Alice*’s Access Control List for S . Let R be the set of privileges needed for accessing S . The problem is: when *Bob* submitting a request r through a proxy, say P , to access resource S , how can *Alice* be sure that the request shall be granted the permission, i.e., *Alice believes* r where $r \in R$. The related credentials in the scenario are shown as following:

- $C1 : [CA1 \succ_{id\_key} CA2, CA2 \succ_{k_{CA2}} k_{CA1}]_{k_{CA1}^{-1}}$ (cross certificate to $CA2$)
- $C2 : [Bob \succ_{k_{Bob}}, CA2 \succ_{id\_key.pc} Bob]_{k_{CA2}^{-1}}$ (identity certificate of *Bob*)
- $C3 : [P \succ_{k_P}, Bob \succ_R P, Bob \succ_{id\_key.pc} P]_{k_{Bob}^{-1}}$ (proxy certificate to P)
- $C4 : [r]_{k_P^{-1}}$ (request certificate from proxy P)

The procedure for compliance checking from *Alice*’s view are shown as following:

- (1) *Alice* $\succ_{id\_key} CA1$ (from *Alice*’s local policy)
- (2) $CA1 \succ_{k_{CA1}}$ (from *Alice*’s local policy)
- (3) *Alice* $\succ_R Bob$ (from *Alice*’s local policy)
- (4) k_{CA1} believes $CA2 \succ_{k_{CA2}}$ (from cross certificate $C1$)
- (5) k_{CA1} believes $CA1 \succ_{id\_key} CA2$ (from cross certificate $C1$)
- (6) $CA1$ believes $CA2 \succ_{k_{CA2}}$ (from (2) and (4) by rule $IR1$)
- (7) $CA1$ believes $CA1 \succ_{id\_key} CA2$ (from (2) and (5) by rule $IR1$)
- (8) $CA2 \succ_{k_{CA2}}$ (from (1) and (7) by rule $IR1$)
- (9) k_{CA2} believes $Bob \succ_{k_{Bob}}$ (from identity certificate $C2$)
- (10) k_{CA2} believes $CA2 \succ_{id\_key.pc} Bob$ (from identity certificate $C2$)
- (11) $CA2$ believes $Bob \succ_{k_{Bob}}$ (from (8) and (9) by rule $IR1$)
- (12) $CA2$ believes $CA2 \succ_{id\_key.pc} Bob$ (from (8) and (10) by rule $IR1$)
- (13) $CA1 \succ_{id\_key} CA2$ (from (1) and (7) by rule $IR5$)
- (14) *Alice* $\succ_{id\_key} CA2$ (from (1) and (13) by rule $IR3$)
- (15) $CA2 \succ_{id\_key.pc} Bob$ (from (12) and (14) by rule $IR5$)

(16) $Alice \succ_{id\_key.pc} Bob$	(from (14) and (15) by rule <i>IR3</i>)
(17) $Bob \succ k_{Bob}$	(from (11) and (14) by rule <i>IR1</i>)
(18) $k_{Bob} believes P \succ k_P$	(from certificate <i>C3</i>)
(19) $k_{Bob} believes Bob \succ_R P$	(from certificate <i>C3</i>)
(20) $Bob believes P \succ k_P$	(from (17) and (18) by rule <i>IR1</i>)
(21) $Bob believes Bob \succ_R P$	(from (17) and (19) by rule <i>IR1</i>)
(22) $P \succ k_P$	(from (16) and (20) by rule <i>IR1</i>)
(23) $Bob \succ_R P$	(from (3) and (21) by <i>IR5</i>)
(24) $Alice \succ_R P$	(from (3) and (23) by <i>IR3</i>)
(25) $k_P believes r$	(from certificate <i>C4</i>)
(26) $P believes r$	(from (22) and (24) by <i>IR1</i>)
(27) $Alice believes r$	(from (24) and (26) by rule <i>IR1</i>)

5 Related Works

In recent years many access control systems, including PolicyMaker [6], KeyNote [7], SPKI/SDSI [17], and SRC [11], have been proposed. The PolicyMaker, and KeyNote focus on expressing security policies with a language, and enforcing them in decentralized environments. The SPKI/SDSI is reaction to the perceived complexity of X.509. It binds permissions directly to keys, and focuses on permission delegation. The SRC system is conceptually similar to our logic. Both of the two logics, Lampson's and ours, reason about security issues based on modal logic and Kripke semantics using the relationships between entities. The major difference between the two logic is that our logic uses fine-grained trust relations between entities to encode credentials rather than the partial order "speaks-for" relation in Lampson's logic. So our logic is able to represent finer privilege delegation in distributed systems whereas in Lampson's logic, a restricted privilege delegation only can be represented by introducing an additional "role" [10].

6 Conclusion and Future Works

In this paper, we propose a formal logic that can be use to reason about access control mechanism in the open, large-scale distributed system, i.e, the grid. Our logic has a strong ability to encode credentials and security policies. It captures some inherent properties of the security problems in the grid, and can be used as theoretical basis for trust engine that plays an important role in decision making for requests to shared resources in the grid environment.

By now we are implementing a general trust engine based on our logic in the GSI [2] framework. We are intended to enhance the GSI so that it supports the features such as PKI interoperability and Role-based authorization delegation.

References

1. Foster, I., Kesselman, I. (ed.): The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann, San Francisco (1999)

2. Foster, I., Kesselman, C., Tuecke, S.: A security architecture for computational grid. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, (1998) 83–92
3. Weeks, S.: Understanding trust management systems. *Proc. of the IEEE Symp. on Security and Privacy* (2001) 94–105
4. Thompson, M., Olson, D., Cowles, R., Mullen, S., Helm M.: CA-based trust model for grid authentication and identity delegation. *Grid Certificate Policy WG. Global Grid Forum* (2002)
5. Steve, L., David, F., R., Richard, L., Steve, O., John, W.: CA-CA Interoperability. *White Paper PKI Forum* (2001)
6. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. *Proc. of the Symp. on Security and Privacy* (1996) 64–173
7. Blaze, M. Feigenbaum, F., Ionnidis, J., Angelos D. K.: The role of trust management in distributed systems security. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, Springer Verlag (1999) 185–210
8. Blaze, M., Feigenbaum, J., Strauss, M.: Compliance checking in the PolicyMaker trust management system. *Lecture Notes in Computer Science*, Vol. **1465**. Berlin Springer (1998) 251–265
9. Halpern, J., Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, **54** (1992) 319–379
10. Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems* **15(4)** (1993) 706–734.
11. Lampson, B., Abadi, M., Burrows, M., Wobber, E.: Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems*, Vol. **10(4)** (1992) 256–310
12. Howell, J., Kotz, D.: A formal semantics for SPKI. *Technical Report TR2000-363*, Dartmouth College (2000)
13. Rangan, P.: Modeling trust management system for grids. *Lecture Notes in Computer Science*, Vol. 3032. Springer-Verlag, Berlin Heidelberg (2004) 899–906
14. Rangan, P.: An axiomatic basis of trust in distributed systems. *Proc. of the IEEE CS Symp. on Research in Security and Privacy* (1988) 204–211.
15. Grandison, T., Sloman, M.: A survey of trust in Internet applications. *IEEE Communications Surveys*, Vol. **4(4)** (2000) 2–16
16. Tuecke, S., Engert, D., Foster, I., Thompson, M., Pearlman, L., Kesselman, C.: Internet X.509 public key infrastructure-proxy certificate profile. *Internet Draft* (2001)
17. Carl, M.E., Bill, F., Butler, L., Ron, R., Brian, M.T., Tatu, Y.: SPKI certificate theory. *Internet RFC 2693* (1999)
18. Andrew W.A., Edward, W.F.: Proof-carrying authentication. *6th ACM Conference on Computer and Communications Security*, Singapore (1999) 52–62
19. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. *IEEE Computer* **29(2)**, IEEE Press (1996) 38–47

Security Enhanced to GSI: An Integrated Framework with a Mechanism

Baoliang Zhang¹, Hanping Hu¹, Xiaogang Wu¹, and Tao Kong²

¹ State Education Ministry Key Laboratory of Image Information & Intelligent control, Institute of Pattern Recognition & Artificial Intelligence
Huazhong University of Science & Technology, Wuhan 430074, Hubei, China
sacc2000@sina.com

² 709th Research Institute of CSIC, Wuhan 430074, Hubei, China

Abstract. This paper presents a framework for GSI to integrate security components from the viewpoint of connection. To enhance security and security of virtual organization, A multi-homed architecture model is given, which employs a mechanism based on modified SOCKSv5 to support the resource utilization for GSI itself. The framework and the modified mechanism is independently. With the mechanism, the GSI can provide dynamic connection through multiple exits in a virtual organization transparently, and the traffic flow can switch smoothly among the multiple proxies by maintaining a coherent connection context. With the cooperation of some other GSI security components, the model and the mechanism enhance the security and performance of GSI to some extent.

1 Introduction

The GSI uses public key cryptography as the basis for its functionality, then employs some security mechanisms, such as digital signature, certificates, mutual authentication, confidential communication, delegation and single sign-on to allow users and applications to securely access Grid resource, and it support standardized APIs such as GSS-API [1, 2]. But there exists many risks in GSI as well, for example, DOS attacks can run out of network resource and blocking the communication. Firewall, private IP addresses etc. sometimes hamper connectivity [3]. With the more and more application of Grids, the limited bandwidth of a virtual organization will not meet the requirement of many users or applications. Therefore, The GSI should be enhanced to provide availability and reliability through extension to the basic mechanism concerning security and performance.

Some protocols, such as SSL and IPSec, are more lightweight without over-provisioning give no assured quality of service at all, and the reliability of the network. The lack of guaranteed resource is a major factor [4]. So multi-homed cluster system can be employed to improve system performance, since the traffic flow can be switched on demand, otherwise the over-provisioning resources will not be exploited sufficiently [5]. Therefore, transmission switch is a key mechanism.

To improve the system reliability and performance, some cluster systems employ multiple different software components to implement parallel processing, or split the

protocol operation into pipeline, such as TCP Router [6] and LVS [7]. In [8], a mobile communication mechanism, MSOCK+, gives the idea of split transmission is introduced to maintain the integrity of connection on transport layer, similar to [3]. However, these methods only permit mobile nodes to change parameters related to transmission while the proxy gateway is fixed, which is unsuitable for multi-homed architecture and vulnerable to attacks.

Aiming at the integrated framework and the dynamic switch problem of the multi-homed architecture, the paper proposes a solution. The rest of the paper is structured as follows: Section 2 presents an integrated Framework model, and a multi-homed architecture is proposed in Section 3. Section 4 introduces a dynamic security connection mechanism for the architecture in Section 2. Section 5 gives the experimental results and analysis of the mechanism, and concludes finally.

2 An Integrated Framework Model

In order to support scalable, distributed virtual organization, Grid security model drives the need for multiple security mechanisms. To address this problem, we can use transport connection as basic abstraction of the communication between individual processes, although [3] proposes an integrated solution to performance and security problems based on the concept of “connection”, it does not give a basic framework to accommodate the components.

As well known, GSI is a set of protocols, libraries, and tools that allow users and applications to securely access Grid resource. The Globus Toolkit's implementation of the GSI adheres to the Generic Security Service API (GSS-API), which is a standard API for security systems promoted by the Internet Engineering Task Force (IETF). GSS-API provides only an abstract interface provides security services for use in distributed applications, isolates callers from specific security mechanisms and implementations. The idea of GSS-API considers the scalability and portability, but it has not a framework protocol easy to integrate the security components, even the GSS-API itself. So we select a protocol framework based on connections for the integrated model, which can accommodate many security mechanisms in GSI.

SOCKSv5 is an evolving industry-standard flow-specific proxy protocol designed to allow secure and managed access to external networks by managing the information passing in or out of any session routed through the proxies [9]. This protocol framework accepts different authentication methods and encryption technologies and is used to build firewall. The most traditional and common use of SOCKS is a network firewall, even though SOCKS is much more than just a firewall. It can combine with other technology to provide more service. For example, the combination of SOCKS and SSL/TLS in the transport layer to construct VPN system can provide an economic and practical design choice, and the two protocols can complement in authentication and encryption [11,12]. Therefore, we proposed a transmission switch mechanism for multi-homed cluster system based on SOCKSv5 and TLS to build a lightweight dynamic VPN model [12]. So the mechanism can be replanted under GSS-API Authentication Method for SOCKSv5 given in [2].

So we propose an integrated framework model based on basic connection using SOCKSv5 with GSS-API, the operations of this model is the basic command set

about connection based on SOCKS, but in fact, the integrated model maintains a security context. This model can adapt to the characteristics of the current network and new mechanisms in [3,12]. The model is illustrated in Fig.1. On different level, inside the virtual organization and between virtual organizations, the GSI components maintained unified security contexts (dash line), the operations between processes is under the control of different protocols and security context (solid line). For example, credential and certificates are part of security context. And the trust can be built between virtual organizations. Different from GSI, we add the SOCKS proxy to GSI definitely, for it accords the principles of network design and the current network.

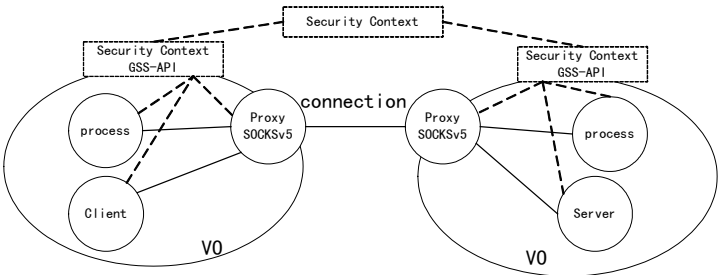


Fig. 1. Integrated Framework Model based on SOCKS

3 A Multi-homed Architecture Model

Based on the above framework model, a multi-homed architecture is given in Fig.2. In order to provide the guarantee of security and QoS to some extent, the proxy cluster is composed of proxy agents in a virtual organization. The proxy cluster and other systems, such as authentication systems, local CA, maintain the security context. When detecting attack behaviors or performance demotion, some traffic flows on the proxy agent should be switched to another agent automatically to keep load balance and prevent attacks. During the switch, the transmission continuity must be guaranteed and the switch overhead should be tolerable. The proxy agent is based on SOCKSv5, so the authentication method and SSL in GSI can be embedded and incorporated in the framework of SOCKSv5 [12]. Therefore, we focus on the switch mechanism from a popular firewall environment to GSS API and GSI components.

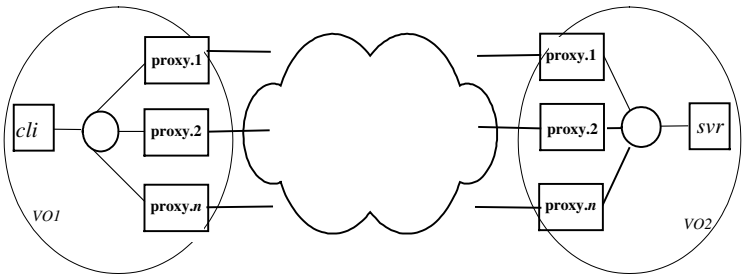


Fig. 2. Multi-homed VPN Architecture

In order to maintain the concurrency, a unified security connection context is maintained among multiple proxies, it is part of the security context of GSI, *cli* is a application deployed in *vo1*, and *svr* is a server application in *vo2*, *proxybox* is a middleware component in the host *cli* stays in, which is a "shim-layer" between the application layer and the transport layer. The *proxy.x* is the SOCKSv5 modules of proxy servers in *vo1* and *vo2*. If *cli* in *vo1* connects to *svr* in *vo2*, the connection is *vo1.cli-vo2.svr*; In fact, this connection is composed of multiple sub-connections by the relay proxies, the connection is *vo1.cli-vo1.proxybox-vo1.proxy.x-vo2.proxy.x-vo2.svr*. The transmission behavior can be customized along the relay path under the condition of maintaining the logic connection *vo1.cli-vo2.svr*, these transmission entities can switch some traffic flows to another path available. The transmission relay modules, *proxybox* and *proxy.x* in different virtual organizations are the switch points, which are the very points to adjust the transmission behavior.

4 A Connection Switch Mechanism

During the transfer of *cli* and *svr* through multi-homed proxies, the traffic flow needs to traverse some proxies and the middleware module. The SOCKSv5 commands support crossing multiple proxies [12]. In GSI infrastructure of VO, the establishment of GSS-API security context can be inserted in the phase "negotiation & authentication" similar to [12].

To build a connection, *cli* invokes socket API function *connect()*; *proxybox* intercepts this function call and makes decision whether the connection request can be submitted to *vo1.proxy.x*. if this transmit admitted, the NMTHMOD fields indicates the GSS-API functions are called to establish the security context, so the mutual authentication, key exchange, confidential communication is determined between the relay modules. Then the CONNECT command of SOCKSv5 is used to build connection, and *vo2.proxy.x* issues request to *svr* for TCP connection. After success, client application receives the successful return of socket function *connect()*. The data exchange between client and server applications is also a relay procedure of confidentiality.

Although the relay transmission can be achieved, the transmission switch cannot be achieved by the commands of SOCKSv5. So the new mechanism we proposed introduces the concept of security connection context for GSI, which is used to manage the transmission switch among multiple proxies. The security connection context is extension to *security context* of GSS API. It is a list containing some connection items represents the status of a connection. It is defined as *SEC_CONN_ITEM*: <*ID*, *srcIP*, *srcPort*, *destIP*, *destPort*, *STATUS_DATA*>. Different from [12], *STATUS_DATA* is a data structure concerning a *security connection* of GSS API, the received or sent characters, and security parameters of TLS and the part of credential. A unified security context for transmission switch in the end system consists of these connection items. The analysis of SOCKSv5 command and modification for transmission switch are given as follows:

During the transmission of the *cli* and *svr* in the multi-homed system, every relay entity maintains a couple of sub-connections. The sub-connection pairs maintained in

the system are: $\langle vol.cli, vol.proxybox \rangle$, $\langle vol.proxybox, vol.proxy.x \rangle$, $\langle vol.proxy.x, vol.proxy.y \rangle$, $\langle vol.proxy.y, vol.svr \rangle$.

Assuming to switch a transmission connection in *proxy.1* to *proxy.2* in *vol*, the sub-connections in the transmission entities *proxybox*, *vol.proxy.x* and *vol.proxy.y* should be altered. Assuming $x=1$ and $y=1$, and four items, *sec_conn_item1* to *sec_conn_item4*, are defined for the above sub-connections respectively, and *sec_conn_item.x*, *sec_conn_item.y* are defined as two variables of connection item.

Considering the different causes of switch, the switch procedure has two steps: request of switch, request of setup connection of switch, the corresponding extended commands are *E_SWITCH* and *E_RECONNECT* respectively, the commands is given by modifying the SOCKSv5 command, and extended commands is shown as:

- *VER* protocol version: X'05'
- *CMD*
 - *E_SWITCH* X'05
 - *E_RECONNECT* X'06
 - *E_SWITCHBIND* X'07
 - *E_REBIND* X'08
- *RSV* RESERVED
- *ATYP* SOCKSv5 compatible
- *DST.ADDR* Variable, SOCKSv5 compatible
- *DST.PORT* X'0000, SOCKSv5 compatible
- *SEC_CONN_ITEM1* security connection item
- *SEC_CONN_ITEM2* security connection item

Unlike the original commands of SOCKSv5, the new fields of *SEC_CONN_ITEM1* and *SEC_CONN_ITEM2* are used for transmission switch of GSI. The field *CMD* is added with some new values. During the switch, the value of field *DST.ADDR* and *DST.PORT* is *NULL*, because the parameters for switch is contained in the fields of *SEC_CONN_ITEM1* and *SEC_CONN_ITEM2*, for example, the value of port, the address and the connection status parameters etc. SOCKSv5 protocol specifies that the value X'09 to X'FF of the field *REP* in the reply command is unassigned, so the mechanism here adds some reply commands. In the same way as the extension of request command, so the reply commands can to match the modification of request commands to implement the transmission switch, it is not given here in detail. As a matter of convenience, and the reply command *E_REPLY_CONN(sec_conn_item.x, sec_conn_item.y)* is used to represent the response of extended request command overall, and the meaning of parameters is not given no longer.

Therefore, the command of switch request is *E_SWITCH (sec_conn_item.x, sec_conn_item.y)*, *sec_conn_item.x* and *sec_conn_item.y* represent the connection items to be switched. The switch connection setup command is: *E_RECONNECT(sec_conn_item.x, sec_conn_item.y)*. The command is *E_REPLY_CONN(sec_conn_item.x, sec_conn_item.y)*. The “*REP*” field is X'0A. The switch procedure of *vol.proxy.1* to *vol.proxy.2* is given below.

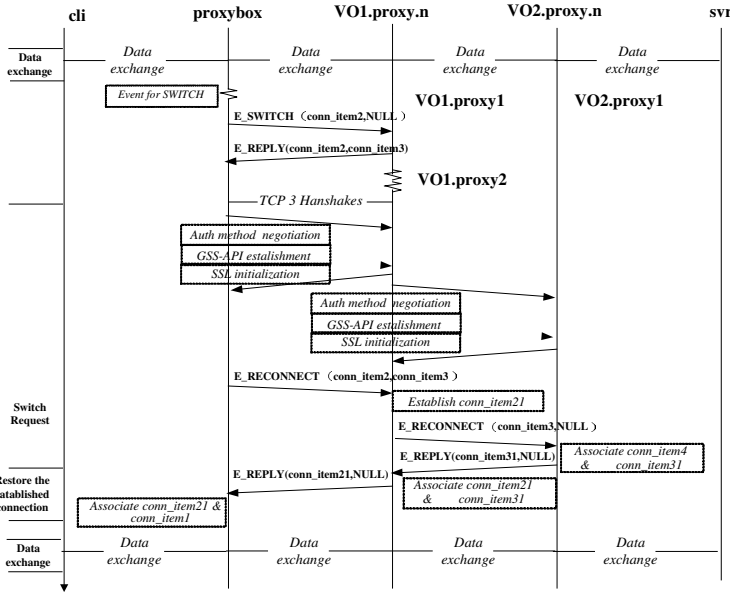


Fig. 3. The Procedure of Dynamic Connection

1. *vo1.proxybox* issues *E_SWITCH(sec_conn_item2,NULL)* to *vo1.proxy1*, *vo1.proxy1* uses the command *E_REPLY(sec_conn_item2, sec_conn_item3)* to return the connection item, and disconnects the origin connection.
2. *vo1.proxybox* sets up a connection with *vo1.proxy2* by 3 way TCP shakes, then the issues command of *E_RECONNECT(sec_conn_item2, sec_conn_item3)*, and negotiates, authenticates and waits for the return result.
3. *vo1.proxy.2* issues reconnection request to *vo2.proxy.1* by *E_RECONNECT(sec_conn_item2,conn_item3)*, the latter queries the SOCKS connection list, if there exists no *sec_conn_item2*, then sets up a new connection item *sec_conn_item21*, and issues a connection request to *vo2.proxy.1* by *E_RECONNECT(sec_conn_item3, NULL)*, *vo2.proxy.1* searches *conn_item3*, set up a new connection *sec_conn_item31* by the address, port, and ID of *vo1.proxy.2*, and associates with *sec_conn_item4*. Finally, issues response to *vo1.proxy.2* by *E_REPLY(sec_conn_item31,NULL)*.
4. *vo1.proxy.2* issues response to the middleware component in *cli*'s host *vo1.proxybox* by *E_REPLY(sec_conn_item21, sec_conn_item31)*.

Now the new connection is *sec_conn_item1- sec_conn_item21- sec_conn_item31- sec_conn_item4*. The above switch flowchart is illustrated in Figure 4.

This system also defines the commands *E_SWITCHBIND* and *E_REBIND* for the switch of connection set up by the *BIND* command. The corresponding reply command is *E_REPLY_BIND*. Since the transmission flow to be intercepted and relayed is on the session layer, this mechanism need to control the status of session to be switched. The status includes the numbers of octet to be sent or received, the content of credential. The analysis and specification of transmit status assignment of connection items is given in [8].

5 Experiments and Analysis

The simulation environment is built with XBone [13], in our lab LAN. This configuration employs two overlay networks *overlay1* and *overlay2* as two ISP networks, and ERD *host1*, *host2* as proxy gateways at the end of *overlay1*, *host11* and *host21* as proxy gateways at the end of *overlay2*, and host *A* running *proxybox* and host *B* act as the host in end system *vo1* and *vo2*.

In the test environment, and some files can be transfer from A to B, and the correctness of the extension to GSI is verified, and the basic performance is illustrated in Figure 6, so the overhead of switch is tolerable, the switch overhead is relatively small and has little influence on the performance.

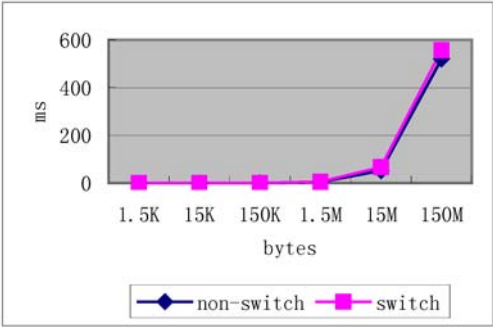


Fig. 4. The Switch Overhead

6 Conclusion

Now the security researches of GSI focus on service security. An integrated model is given based on GSS-API and SOCKS. The new model can employ many new connection mechanisms to improve performance and reliability, which roots in the network design and the current network architecture.

Based on the above model, the multi-homed architecture can meet the demand of the increasing traffic, and prevent single point failure and DOS attack. The session-switch mechanism of multi-homed proxies can switch the transmission smoothly to prevent the traffic analysis and DoS attack. In the paper, the switch mechanism is extract form the framework for simplicity. The Framework model and the mechanism are independent, and the mechanism can be used in some scenarios. The further work is to embedded in the GSI with other components.

Acknowledgments

We would like to thank the support of the National Nature Science Foundation of China (No.99104029). Special thanks to Mr. Liang Xin and Mr. Wang Haibo for their building the simulation environment.

References

1. Jeffrey Dwoskin¹, Sujoy Basu, Vanish Talwar, Raj Kumar, Fred Kitson, Ruby Lee, Security Issues for Interactive Grids, the 37th Annual Asilomar Conference on Signals, Systems, and Computers, 9-12, November 2003, Pacific Grove, California
2. P. McMahon, GSS-API Authentication Method for SOCKS Version 5, IETF RFC1961, June 1996
3. Alexandre Denis, Olivier Aumage, Rutger Hofman, Kees Verstoep, Thilo Kielmann, Henri E. Bal, Wide-Area Communication for Grids: An Integrated Solution to Connectivity, Performance and Security Problems, <http://www.cs.vu.nl/ibis/>
4. Rebecca Isaacs, Ian Leslie, Support for Resource-Assured and Dynamic Virtual Private Networks, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 19, NO. 3, MARCH 2001
5. John Bartlett, Route control-optimized multi-homed connections for Performance, Load and Cost, Business Communication. January, 2002, <http://www.bcr.com>
6. <http://www.research.ibm.com/webvideo/tcproute.html>
7. <http://www.LinuxVirtualServer.org>
8. Pravin Bhagwat, David A. Maltz et al. MSOCKS+: an Architecture for Transport Layer Mobility Computer Networks. 39 (2002) 385-403
9. M. Leech, M. Ganis, and Y. Lee, et al. SOCKS Protocol version 5, RFC 1928, March 1996
10. T. Dierks, C. Allen The TLS protocol version 1 IETF RFC2246, Jan. 1999
11. Yang Chun, The Study and implementation on employing SOCKSv5/TLS to Construct VPN. Journal of University of Electronics Science and Technology of China VOL.30 No.3 June (2001) (in Chinese)
12. Hu Hanping, Zhang Baoliang, Chen xiang and Zhu Haiyan, A Transmission Cluster model based on Active Defense. Proceeding of Conference 2002 on Network and Information Security, Jan. (2003) Beijing, pp.346-353 (in Chinese)
13. Joe Touch, Dynamic Internet Overlay Deployment and Management Using the XBone. Computer Networks, July (2001), pp. 117-135

Reliable Accounting in Grid Economic Transactions

Luigi Catuogno¹, Pompeo Faruolo¹,
Umberto Ferraro Petrillo¹, and Ivan Visconti²

¹ Dipartimento di Informatica e Applicazioni, Università degli Studi di Salerno
Via S. Allende - 84081 Baronissi (SA), Italy
{luicat,pomfar,umbfer}@dia.unisa.it

² Département d'Informatique, École Normale Supérieure
45, rue d'Ulm, 75230 Paris Cedex 05
ivan.visconti@ens.fr

Abstract. In the Grid computing model a remote service is provided by a resource owner to a client. The resource owner executes a client job and charges the client for a corresponding fee. In this paper we discuss the main weakness of many existing models for performing such a kind of transaction, i.e., the strong assumption that both the resource owner and the clients are honest. Then, we propose a new security model in which either the resource owners or the clients (or both) may not be honest. Our model introduces a trusted third party, referred to as “Grid Manager”. We describe in details the role of the Grid Manager and argue the advantages of our proposal with respect to the current state-of-the art.

1 Introduction

Recently a new model of distributed computing referred to as “Grid computing” is emerging. In this model several users share their computational, storage and communication resources for making a global Grid environment [1]. Client users are interested in accessing these resources, therefore they locate the resource providers that better match their requirements and assign them the jobs to be executed, all with the help of the Grid infrastructure. Grid computing has been initially conceived as a way, for the scientific community, to execute computationally intensive jobs. Nowadays, the Grid computing is rapidly evolving as to become a business opportunity in which the actors share their resources in order to make profit. This trend has motivated the development of economic models aiming at defining rules for pricing, trading and charging for services provided by a Grid. A simple approach that is commonly used to this end is to charge the jobs executed on a Grid according to the amount of the resources that they consume. This approach requires the installation of an accounting system in order to somehow measure the resources consumed by a job during its life span, and to translate this measurement into a chargeable price.

The existing accounting systems rely on the assumption that all parties of a Grid economic transaction are honest. We observe that this assumption is unrealistic. Indeed, both parties of a transaction may cheat in order to increase their profit. On one hand, the resource owner could pretend to be paid for an amount of resources greater than the one that he has actually used for the fulfillment of a job. On the other hand, a client could refuse to pay for a service by claiming that he has been cheated out.

In this paper, we propose a new model for building a reliable accounting system. Our model requires the existence of a trusted party in the Grid that guarantees the execution of reliable Grid economic transactions, even in case that clients and resource owners are corrupted. This authority has a trusted and private computing infrastructure that can be used to verify the amount of resources needed to execute a job and to compare this information with the one claimed by a resource owner upon the execution of the same job. The private infrastructure is also used for performing a periodical verification of the behavior of the resource owners in order to discover potential frauds.

This paper is organized as follows. In Section 2 we introduce Grid economic transactions and we briefly review the existing Grid accounting systems. In Section 3 we discuss some of the security issues that arise when implementing Grid economic transactions. In particular, we discuss some possible tasks that a corrupted resource owner can perform in order to fraud a user, by cheating on the cost of a job. Finally, in Section 4 we present and analyze our model for performing reliable accounting on the Grid.

2 Grid Economic Transactions

In a typical Grid economic transaction we distinguish two parties: a resource owner R , that joins a Grid with his hardware and software infrastructure, and a client C , that asks the Grid for the execution of a job J . The aim of R is to make a profit by providing his infrastructure for the execution of client jobs. The aim of C is to execute the job J without having the corresponding hardware and software infrastructure; thus she pays a fee for obtaining such a service from the Grid. The interaction between these two parties is mediated by the Grid infrastructure. This party is a broker that offers to the users the services needed for discovering, choosing and accessing the resources that best match their requirements. The execution cost of J is determined according to some quantitative (e.g., the total amount of resources required for the execution of J) or qualitative (e.g., the computational power of the server) metrics.

The implementation of Grid transactions where clients are charged according to the resource consumption of their jobs, requires the introduction of an accounting system for measuring and collecting the resource usage data of the jobs executed on the Grid.

The Open Grid Service Architecture (OGSA) [1], currently the *de facto* standard for the implementation of Grids, includes an accounting subsystem composed by several services to be used as building blocks for developing a “Grid

economy”. The *metering service* is used to measure the resource usage of a job. The *rating service* concerns the translation of data about consumed resources into chargeable prices. The *accounting service* charges a specific user for the cost computed by the rating service. The *billing service* interacts with some external financial services in order to manage users payments.

The current state-of-the art presents several accounting systems as the Grid Service Accounting Extensions [2], the Grid Economic Services Architecture [3], SNUPI [4] and GridBank [5]. In such systems, each grid node runs a “monitor agent”, a process that measures the resource usage of every job executed by the node. Measurements are accomplished by means of the operating system accounting facilities or through the profiling features of the employed real-time environment (*e.g.*, the Java virtual machine). The agents send the collected information to a trusted third-party that manages the accounting process.

3 Security Issued in Grid Economic Transactions

The use of any of the existing accounting systems in the fulfillment of a Grid economic transaction brings up several security issues. Consider the following metaphor. When a person buys some fruits she can verify by herself their weight and, thus, she is able to trivially evaluate the corresponding total cost. Such a verification cannot be performed in Grid transactions when the cost of a job depends on the resources it requires. Indeed, *an user will likely not known in advance the exact amount of resources needed for accomplishing her job*. Moreover, in many cases, she is not able to verify this by herself since she does not have the corresponding hardware and software infrastructure. Thus, the price that an user has to pay is completely due to the amount of resources that the monitor agent, running on the resource owner machine, reports.

The strong assumption that all parties are honest does not correspond to the real context of Grids. Indeed, since the resource owners join a Grid for making a profit, they are strongly motivated in deviating from the specification of the standard protocol in order to increase their profit. For instance, a resource owner can easily cheat by specifying an amount of resources used to execute a job that is different (actually greater) with respect to the real one. In a similar way, a client could refuse to pay for a service by claiming that he has been cheated out. A consequence of such a weakness is that a user pays too much or does not pay at all and thus the quality of the service offered by the Grid decreases.

Cheating an Accounting System. We now discuss some malicious activities of a corrupted resource owner that tries to fraud a user by cheating on the amount of consumed resources for the execution of a job. We observed in Section 2 that the existing accounting systems meter the resource usage of a job by running a monitoring software agent on the machine hosting the job itself. This approach relies on a strong assumption: *the monitor agent trusts the hardware and the operating system it is running on*. Indeed, a malicious resource owner could cheat a monitoring agent that is running on his infrastructure without even modifying

the agent code. This can be done by leveraging the underlying operating system in order to provide incorrect information to the monitoring software, since this information is obtained by querying the hosting operating system.

Another possible strategy for cheating is to corrupt, at run time, the monitoring agent by means of techniques of intrusion, such as [6], in order to deviate its execution. In such a case, the other modules of the accounting system that interact with the monitoring agent do not realize that it has been tampered.

Finally, a malicious resource owner can also cheat by running a corrupted monitoring agent instead of the one distributed by the accounting system.

As it comes out trivially, in these cases, neither the accounting service nor the user that issued the job could be able to detect such a fraud.

4 Secure Grid Transactions

In this section we present our architecture for the execution of secure Grid transactions. We first introduce the model on which we base our architecture, then we describe and analyze the execution of Grid transactions in the proposed model.

4.1 The Model

The accounting and monitoring systems proposed in the past require the existence of a trusted third party (see Section 2). In our model, we follow the same lead of the previous proposals assuming the existence of a trusted third party, in particular we try to exploit the reliability of such a party in order to design secure transactions on the Grid. We refer to the *Grid Manager* (GM), as the interface between clients and resource owners (in Section 2 we referred to such a party as the Grid infrastructure). GM decides which resource of the Grid has to be used in order to satisfy a client request.

Note that the aim of GM is to have as many resource owners as possible in order to execute the jobs of a lot of clients. Therefore GM is interested in protecting both users from corrupted resource owners and resource owners from corrupted users. In order to achieve that the GM has a private computing infrastructure to verify the real amount of resources needed to execute a job. Since GM has an “institutional” role, we assume that it is the only trusted party of our model.

Monitoring. The execution of a transaction in a Grid is a remote service between a client that needs the execution of a job and a resource owner that has the hardware and software resources to execute the job. In the last stage of such a remote service the resource owner charges the client for the amount of resources that he has used for executing the job. Since an honest client simply pays the charged amount, a malicious resource owner could try to cheat by charging the client for resources that he has not spent during the execution of the job.

GM performs the following monitoring activity in order to detect the existence of malicious resource owners in the Grid.

- **GM** maintains a set S of “testing” jobs such that the distribution of the resources needed for their execution is statistically close to the distribution of the resources needed by the jobs submitted by the users.
- **GM** randomly chooses a resource owner and assigns him a job randomly chosen from S . Note that since the resources needed by the jobs chosen from S have the same statistical distribution of the resources needed by the jobs submitted by real clients, the resource owner cannot distinguish a testing job from a real client job. Consequently, in case a malicious resource owner tries to cheat, the monitoring of **GM** detects such a malicious activity.

Note that the trade-off between testing jobs and real client jobs defines a quality metric of the Grid.

Fraud Verification. The monitoring of **GM** is not a catch-all solution with respect to malicious resource owners. In particular, in order to preserve the performance of the Grid, the workload of the monitoring must be bounded by a percentage of the overall workload.

The aim of this procedure is to detect malicious resource owners that are not discovered by the monitoring. Fraud verification is a procedure invoked from a client that feels cheated. In this case **GM** executes the job on his private infrastructure in order to verify whether the client has been fraud by the resource owner.

4.2 The Architecture

In this section we describe our proposal for the execution of reliable transactions in the Grid computing model introduced above.

Set-Up of the System. **GM** generates a pair $(\text{pk}_{GM}, \text{sk}_{GM})$ respectively of public and private keys for a secure digital signature scheme. We suggest to use the RSA encryption scheme implemented with the optimal asymmetric encryption padding. Such scheme has been proved to be secure (in the adaptive chosen ciphertext attack sense [7]) in [8] considering the random oracle model [9]. Moreover, **GM** chooses a function h from a family of collision resistant hash functions.

We assume that **GM** possesses an heterogeneous hardware and software infrastructure. Such an infrastructure is composed by a minimal set of heterogeneous workstations that can be used to measure the amount of resources needed by any job executed in the Grid. Moreover we assume that **GM** possesses a database in which he can log the transcripts of the transactions performed in the Grid. After the set-up of the system, **GM** will play also the role of certification authority.

User Enrollment. The enrollment is a procedure performed by **GM** along with a client or a resource owner.

- **Client enrollment:** The client performs such a procedure in order to obtain the privileges for accessing the Grid. The client generates a key pair $(\text{pk}_c, \text{sk}_c)$ (with the same requirements described in the set-up) and asks for a digital

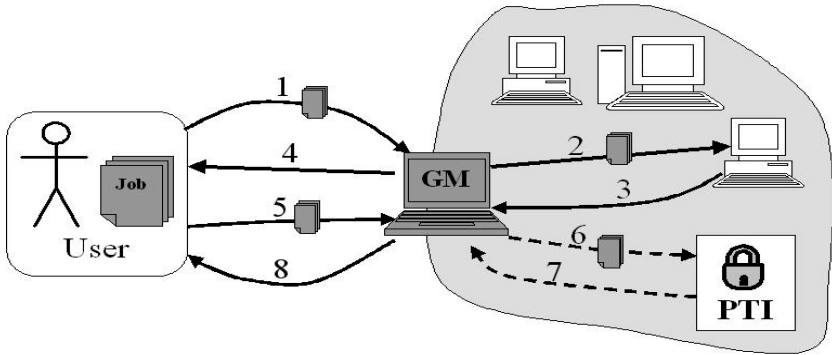
certificate. **GM** verifies the identity of the client and uses his secret key $\mathbf{sk}_{GM}$ to compute a standard digital certificate (X509v.3 [10]) corresponding to the identity of the client and to his public key $\mathbf{pk}_c$.

- **Resource owner enrollment:** The resource owner performs such a procedure in order to make his hardware and software infrastructure available to clients. The resource owner generates a key pair $(\mathbf{pk}_r, \mathbf{sk}_r)$ (with the same requirement described in the set-up) and asks for a digital certificate. **GM** verifies the identity of the resource owner (optionally, **GM** could also verify the hardware and software resources). Finally, as in the previous case, **GM** computes a corresponding digital certificate but in this case the public key encoded is $\mathbf{pk}_r$.

Execution of a Transaction. The execution of a transaction, depicted in Fig. 1, is a procedure in which all the three possible parties are involved: the client C , the Grid manager **GM** and the resource owner R . We distinguish the following steps during the execution of this procedure.

- C submits a job J . He generates a random serial number s_c and uses his secret key $\mathbf{sk}_c$ to compute a digital signature $\hat{J}_c$ of the pair (J, s_c) . C sends the triplet $(J, s_c, \hat{J}_c)$ to **GM**.
- **GM** verifies that $\hat{J}_c$ is a valid signature of (J, s_c) with respect to the public key $\mathbf{pk}_c$ and that s_c has never been received in the past from C . Then **GM** computes $H_J = h(J)$ and stores the triplet $(H_J, s_c, \hat{J}_c)$ in his database. Note that the size of the triplet is constant and independent of the size of the job J . Then **GM** generates a random serial number s_{GM} and uses $\mathbf{sk}_{GM}$ to compute a signature $\hat{H}_J$ of (H_J, s_{GM}) , chooses a resource owner R among the available resource owners and sends him $(J, s_{GM}, \hat{H}_J)$.
- R verifies that $\hat{H}_J$ is a valid signature of (H_J, s_{GM}) with respect to the public key of **GM** and that he has not received in the past the same serial s_{GM} from **GM**. Then R executes the job J and measures the resources needed during the execution. R generates a random serial number s_r and uses his secret key $\mathbf{sk}_R$ to compute a signature $\hat{I}_R$ of a digital invoice I_R that includes (H_J, s_r, s_{GM}) and a description of the used resources along with their corresponding fee. The digital invoice I_R and the signature $\hat{I}_R$ are sent to **GM**.
- **GM** verifies that $\hat{I}_R$ is a valid signature of I_R with respect to the public key of R , that the invoice refers to a job previously sent by **GM** to R and that no other invoice has been sent by R to **GM** with respect to the same job. **GM** adds his fee and uses his secret key $\mathbf{sk}_{GM}$ to compute and sign a new digital invoice I_{GM} that includes I_R . **GM** sends to C such a payment request and updates the database by adding I_{GM} to the previously stored triplet corresponding to J .
- C verifies that the digital invoice is correctly signed by **GM** and that refers to the same job J whose execution he asked for. If C has not received in the past such an invoice, and if the charged amount belongs to given expected range, then he pays **GM** for the charged amount.
- **GM** receives the payment of C and pays R for his corresponding amount.

If the amount specified in the invoice does not belong to the range expected by C , he rejects the invoice and asks for a fraud verification procedure, by sending to **GM** the job J and the serial number s_c previously submitted. **GM** computes again the hash H_J of J and verifies that the same job is referred to in the invoice received from the resource owner. Then **GM** executes J in his private trusted infrastructure in order to measure the resources needed by its execution. If the invoice was correctly computed, **GM** again charges a fee to the user since he has to pay for the use of the private infrastructure. If, instead, the amount specified in the invoice is greater than the measured one, the user is not charged for the execution of J . In both cases, a ranking process, such as the one presented in [11], is run to log these behaviors. The outcoming ranks would then be used to penalize malicious users during the trading phase for the bargaining of new jobs.



1. A user submits a job.	5. The user submits the same job to GM.
2. GM chooses a resource owner.	6. GM executes again the job using his PTI.
3. GM receives resource usage data.	7. GM receives trusted usage data.
4. GM sends an invoice to the user.	8. GM compares the two outputs.

Fig. 1. A sketch of the fraud verification procedure.

Verification Issues. As already discussed, during the fraud verification procedure, **GM** verifies the invoice generated by a resource owner after the execution of a job by running the same job in his private trusted infrastructure.

A first consequence is that **GM** is able to verify only the jobs that can be executed in its private infrastructure. This is not generally an hard problem since the number of operating systems and hardware architectures spanning the most part of existing computing infrastructures is small (e.g., Linux/x86, MacOS/PowerPc, Java). By using these architectures in its private infrastructure, **GM** would be able to support the verification procedure for a large number of cases.

A second consequence is that the infrastructure used by **GM** for verifying a job could have a different performance (e.g., because of a different clock speed or

a larger amount of physical memory) with respect to the infrastructure used by the resource owner. More precisely, the resource usage reported by the different machines running the same job with the same data files could not be comparable. Indeed, there are some resources as the maximum amount of memory to be allocated for the execution of a job that can be measured independently of the overall performance of the system. On the contrary, there are some resources whose measurement strongly depends on the overall performance of the system (e.g., the CPU time assigned to the execution of a job). In this last case we consider two alternatives. The first alternative is to use some *a priori* knowledge about the performance of a machine in order to normalize the reported resource usage. The second alternative is to combine these measurements with some quantitative information able to describe the total amount of work done by a system while processing a job (e.g., considering the total number of assembler instructions issued for the execution of a job).

References

1. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In: Open Grid Service Infrastructure WG, Global Grid Forum (2002)
2. Beardsmore, A.: Grid Service Accounting Extensions (GSAX). <http://www.doc.ic.ac.uk/~sjn5/GGF/ggf-rus-gsax-01.pdf>
3. Grid Economic Services Architecture (GESA). <http://www.doc.ic.ac.uk/~sjn5/GGF/draft-ggf-gesa-services-1.pdf> (2003)
4. Hazlewood, V., Bean, R., Yoshimot, K.: SNUPI: A grid accounting and performance system employing portal services and RDBMS back-end. In: The 5th LCI International Conference on Linux Clusters: The HPC Revolution. (2004) 18–20
5. Barmouta, A., Buyya, R.: GridBank: A Grid Accounting Service Architecture (GASA) for Distributed System Sharing and Integration. In: The 17th International Parallel and Distributed Processing Symposium (IPDPS 2003). (2003) 22–26
6. Cowan, C., Wagle, P., Pu, C., Beattie, S., Walpole, J.: Overflows: Attacks and Defenses for the Vulnerability of the decade. In: DARPA Information Survivability Conference and Expo (DISCEX). Volume 2., IEEE Computer Society Press (2000) 119–129
7. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes. In: Advances in Cryptology - CRYPTO. (1998) 26–45
8. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: Rsa-oaep is secure under the rsa assumption. In: Advances in Cryptology - CRYPTO. (2001) 260–274
9. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: ACM Conference on Computer and Communications Security. (1993) 62–73
10. Housley, R., Ford, W., Polk, W., Solo, D.: Internet X509 Public Key Infrastructure: Certificate and CRL Profile. Network Working Group. (2002) RFC 3280
11. W. Chen, W.Z., Yang, G.: On the Malicious Participants Problem in Computational Grid. In: Grid and Cooperative Computing: Second International Workshop, (GCC 2003). Volume 3032 of Lecture Notes in Computer Science., Springer-Verlag (2004) 839–848

A Gravity-Based Intrusion Detection Method

Sheng-Yi Jiang, Qing-Hua Li, and Hui Wang

Computer School, Huazhong University of Science & Technology
430074, Hubei, wuhan, China

Jiangshengyi@163.com, Lqh@263.net, Wh621004@yahoo.com

Abstract. It is an important issue for the security of network to detect new intrusions attack. We introduce the idea of the law of gravity to clustering analysis, and present a gravity-based clustering algorithm. At the same time, we present a simple method calculating cluster threshold. Based on these, a new intrusion detection method is introduced in this paper. The detection method has the nearly linear time complexity with the size of dataset and the number of attributes, which results in good scalability. The experimental results on dataset KDDCUP99 show that our method outperforms the existing unsupervised intrusion detection methods on accuracy and can detect new intrusions.

1 Introduction

Intrusions pose a serious security threat in network environment, and therefore need to be promptly detected and dealt with. Various techniques for modeling anomalous and normal behavior have been developed for intrusion detection. The signature-based detection methods and supervised anomaly detection methods can only detect previously known intrusion, at the same time signature database and labeled data have to be manually processed.

To solve these difficulties, unsupervised anomaly detection methods have been addressed recently [2-5,7]. These methods attempt to find intrusion buried within the data, and needn't any prior knowledge about training data and new attacks. These methods are based on two basic assumptions about the data. The first assumption is that the number of normal instances vastly outnumbers the number of anomalies. The second one is that data instances with same classification (type of attack or normal) should be close to each other in feature space under some reasonable metric, and instances with different classifications will be far apart.

However, existing unsupervised methods have some shortages as follows: (1) In the course of clustering, only distance between an object and class is taken into account, while the effect of the size of class is not. (2) It isn't reasonable that the objects in the small clusters are labeled anomalous. For example, we hypothesize in figure 1 that C1 includes 1000 objects, C2, 800 objects, C3, 75 objects and C4, 25 objects. If we determine abnormal class by the size of class, then the abnormal degree of C4 is greater than that of C3. But C3 departs from the whole set more than C4, so C3 should be determined as abnormal class firstly.

This paper is mainly concerned with these problems. The main contributions of this paper are as follows:

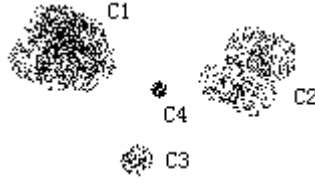


Fig. 1. Relativity between size and outlier of cluster

- We introduce the idea of universal gravity to clustering analysis, and present a gravity-based clustering algorithm. At the same time, we present a simple method calculating cluster threshold.
- We present a concept of gravity factor for cluster, which identifies the degree of a cluster deviating from the whole, and can well distinguish anomalous classes from normal classes.
- We present a novel strategy for detecting intrusion, which achieves both high detection rate and low false alarm rate, and also can detect new intrusions.

The experimental results on dataset KDDCUP99 show that our method outperformed the existing methods on accuracy. (See table 1)

Table 1. The contrast of results with different methods on dataset KDDCUP99

Ref.	Detection rate	False alarm rate	Detection rate for unknown attack
[1]	91.8%	0.5%	/
[3]	28%-93%	0.5%-10%	/
[4]	55%-82%	0.8%-4.9%	/
[5]	43.1%-75.2%	/	/
[7]	35.7%-88%	1.44%-8.14%	/
Our method	93.08%-98.55%	0.52%-2.45%	54.04%-78.92%

Note: “/” denote corresponding data were not given in the literature

The rest of this paper is organized as follows. In section 2, some definitions used in the paper are formalized. Section 3 presents gravity-based clustering method and intrusion detection method. In section 4, we give the methods to select parameters. Experimental results are given in section 5. Finally, section 6 concludes the paper.

2 Definitions

For the convenience of describing, we present five definitions. Supposing dataset D is featured by m attributes (m_c categorical and m_n continuous), D_i is the set of i -th attribute value. For simply, we set categorical attributes before continuous attribute.

Definition 1: For a cluster C and $a_i \in D_i$, then the support of a_i in C with respect to D_i is defined as $Sup_{C|D_i}(a_i) = |\{object | object \in C, object.D_i = a_i\}|$.

Definition 2: For a cluster C , the cluster summary information (CSI) for C is defined as: $CSI = \{kind, n, Summary\}$, where ‘kind’ is the type of the cluster C with value of

‘normal’ or ‘attack’, ‘ n ’ is the size of the cluster C , and ‘*Summary*’ consists of two parts, one describing the frequency information for categorical attribute value, the other describing the centroid of numerical attributes.

$$\text{Summary} = \{ \langle \text{Stat}_i, \text{Cen} \rangle \mid \text{Stat}_i = \{ (a_j, \text{Sup}_{C|D_i}(a_j)) \mid a_j \in D_i \}, 1 \leq i, j \leq m_C, \\ \text{Cen} = (p_{m_C+1}, p_{m_C+2}, \dots, p_{m_C+m_N}) \}$$

Definition 3: For subcluster C, C_1 and C_2 of cluster D , and objects $p = \{p_i \mid i \in [1, m]\}$, $q = \{q_i \mid i \in [1, m]\}$.

(1) The distance between objects p and cluster C , $d(p, C)$ is defined as

$$d(p, C) = \left(\frac{\sum_{i=1}^m \text{dif}(p_i, C \mid D_i)^2}{m} \right)^{1/2}, \text{dif}(p_i, C \mid D_i) \text{ is defined as the distance between objects } p$$

and cluster C on attribute D_i . For categorical attributes, $\text{dif}(p_i, C \mid D_i)$ is defined as

$$\text{dif}(p_i, C \mid D_i) = 1 - \frac{\text{Sup}_{C|D_i}(p_i)}{|C|}, \text{ while for numerical attribute, } \text{dif}(p_i, C \mid D_i) \text{ is defined as} \\ \text{dif}(p_i, C \mid D_i) = |p_i - c_i|.$$

(2) The distance between clusters C_1 and C_2 , $d(C_1, C_2)$ is defined as

$$d(C_1, C_2) = \left(\frac{\sum_{i=1}^m \text{dif}(C_1 \mid D_i, C_2 \mid D_i)^2}{m} \right)^{1/2}, \text{dif}(C_1 \mid D_i, C_2 \mid D_i) \text{ is defined as the distance between}$$

C_1 and C_2 on attribute D_i . For categorical attributes, $\text{dif}(C_1 \mid D_i, C_2 \mid D_i)$ is defined as

$$\text{dif}(C_1 \mid D_i, C_2 \mid D_i) = 1 - \frac{1}{|C_1| \cdot |C_2|} \sum_{p \in C_1} \text{Sup}_{C_1|D_i}(p_i) \cdot \text{Sup}_{C_2|D_i}(p_i) \\ = 1 - \frac{1}{|C_1| \cdot |C_2|} \sum_{q \in C_2} \text{Sup}_{C_1|D_i}(q_i) \cdot \text{Sup}_{C_2|D_i}(q_i)$$

while for numerical attribute, $\text{dif}(C_1 \mid D_i, C_2 \mid D_i)$ is defined as $\text{dif}(C_1 \mid D_i, C_2 \mid D_i) = |c_i^{(1)} - c_i^{(2)}|$.

Definition 4: The gravity between clusters C_1 and C_2 , $g(C_1, C_2)$ is defined as:

$$g(C_1, C_2) = \frac{\sqrt{\ln(C_1 \cdot n + 1) \cdot \ln(C_2 \cdot n + 1)}}{d(C_1, C_2)}, \ln(C \cdot n + 1) \text{ is regarded as the mass of cluster } C.$$

Definition 5: Let $C = \{C_1, C_2, \dots, C_k\}$ be the results of clustering on training data D . The gravity factor of cluster C_i , $GF(C_i)$ is defined as harmonic means of gravities between cluster C_i and other clusters: $GF(C_i) = (k-1) / \sum_{j \neq i} \frac{1}{g(C_i, C_j)}$.

The gravity factor of C_i , $GF(C_i)$ measures how a cluster is attracted by the whole dataset, the less $GF(C_i)$ is, the more outer C_i depart from the whole.

3 The Gravity-Based Intrusion Detecting Method

3.1 Clustering

To create clusters from the input objects, we introduce the idea of gravity to clustering, and present a gravity-based clustering algorithm. We regard the course of clustering as the course which objects are attracted by existing clusters. The detail about the clustering is described as follows.

Step 1: Initializing the set of clusters, S , to the empty set, read a new object p .

Step 2: Creating a cluster with the object p .

Step 3: If no objects are left in the database, turning to step 6, else reading a new object p , finding the cluster C^* in S , such that for all $\bar{C}$ in S ,
 $g(p, C^*) \geq g(p, \bar{C})$.

Step 4: If $g(p, C^*) < r$, turning to step 2.

Step 5: Merging object p into cluster C^* and, modify the CSI of cluster C^* .

Step 6: Stopping.

3.2 The Intrusion Detection Method

We propose a new strategy for intrusion detection, which is composed of modeling and detecting module, the details can be described as follows.

(1) Setting up Model

Step 1, **Clustering:** Cluster on training set T_1 , to produce clusters $C = \{C_1, C_2, \dots, C_k\}$.

Step 2, **Labeling clusters:** Sorting clusters $C = \{C_1, C_2, \dots, C_k\}$ and making them meet

$GF(C_1) \leq GF(C_2) \leq \dots \leq GF(C_k)$. Search the smallest b , which satisfies $\frac{\sum_{i=1}^b |C_i|}{|T_1|} > \varepsilon$, and

then label clusters $C_1, C_2, \dots, C_{b-1}$ with 'attack', while $C_b, C_{b+1}, \dots, C_k$ with 'normal'.

Step 3, **Producing model:** The model consists of the cluster summary information and the threshold r .

(2) Detecting Attack

For any object p in testing set T_2 , find a cluster C_{i_0} that is produce the largest attraction to p , if $g(p, C_{i_0}) \geq r$ then classifies p according to the label of C_{i_0} ; else regard p as new attack.

3.3 Time Complexity

The time complexity of the clustering, the first step of setting up model, depends on the size of training set (N_1), the number of attributes (m), the number of the $CSIs$ and the size of every CSI . To simplify the analysis, we assume the final number of the clusters is k ; categorical attribute D_i consists of distinct values n_i . In the worst case,

we can get that the clustering algorithm has time complexity $O(N_1 \cdot k (\sum_{i=1}^{m_C} n_i + m_N))$. In the course of clustering, the numbers of producing clusters vary from 1 to k by degrees, the number of attribute value increase gradually. So, in practice, the time complexity can be expected to be $O(N_1 \cdot k \cdot m)$. The second step of setting up model, computing the gravity factor of every cluster by computing distance between any pair of clusters, the time complexity is $O(m \cdot k^2)$. Because of $k \ll N_1$, thus in the worst case, time complexity for setting up model is $O(N_1 \cdot k (\sum_{i=1}^{m_C} n_i + m_N))$, and can be expected to be $O(N_1 \cdot k \cdot m)$.

For detecting process, algorithm scan testing set one pass, the time complexity is similarity to the clustering algorithm, and is $O(N_2 \cdot k \cdot (\sum_{i=1}^{m_C} n_i + m_N))$, there N_2 is the size of testing set T_2 .

Thus it can be seen that the time complexity of every module of the detection method are similar, the time complexity is nearly linear with the size of dataset, the number of attributes and the final number of clusters, which make the detection method deserve good scalability.

3.4 Processing Noise

In the course of setting up model, it is possible that some noises are mixed in training set, as threshold r is given reasonable value, these noise will cluster into some spare cluster (the size of cluster is small). We clean up these noises, and then the number of clusters will be decreased and efficiency of detection will be improved.

4 Selecting Parameters

① Selecting Threshold r

The threshold r can influence the quality of clustering and time-efficiency of the algorithm. In order to gain meaningful clustering results, we must choose reasonable threshold r . According to the process of clustering, threshold r should be less than the average gravity of any pair objects. We use sampling technique and develop the strategy to determine threshold. The details are described as follows:

- (1) Choosing randomly N_0 pairs of objects in the dataset D .
- (2) Computing the gravity between each pair objects.
- (3) Computing the average EX of gravity from (2).
- (4) Selecting r in the range $[EX/3, EX/2]$.

② Selecting Parameter ε

ε is the approximation ratio of outlier to whole dataset. As ε increases, the detection ratio will decrease; meanwhile the false alarm rate will go down at the same time. A rule of thumb in statistics is that the proportion of contaminated data in a dataset is usually less than 5% and almost always less than 15%, so we general let ε be about 0.05. If we have prior knowledge on the ratio, we may select ε more accurate.

5 Experimental Results

Detection rate, false alarm rate, detection rate for unknown attack types are used to measure performance of intrusion detection methods. Detection rate is defined as the ratio of detected attack records to total attack records, false alarm rate is defined as the ratio of the normal records which were detected as attack record to total normal records.

We implement algorithm in VC6.0. The dataset used is the KDDCUP99 [6], which contained a wide variety of intrusion simulated in military network environment. The simulated attacks fell in one of the following four categories: DOS, R2L, U2R, and PROBE. There were a total of 22 attack types, 41 attributes (34 continuous and 7 categorical). The whole dataset is too large, generally, the 10% subset that contains all 22-attack types is used to evaluate the performance of algorithm [7]. We divide the subset into two subsets T1, T2. T1 contains 40459 records (96% normal). T2 contains some unknown attacks type in the T1. We set up model on training set T1, and test model on testing set T2. By computing, $EX=31.5$, let $\varepsilon=0.05$. The table 2 shows partial experimental results with distinct r in the range of [11,15].

Table 2. Detection rate for every attack type with distinct r

Attack type	$r=11$	$r=12$	$r=13$	$r=14$	$r=15$
DOS	93.93%	99.16%	99.11%	99.12%	99.13%
PROBE	38.21%	61.59%	63.33%	64.11%	64.55%
R2L	0.27%	28.33%	28.78%	1.70%	2.06%
U2R	0.00%	0.00%	1.96%	7.84%	17.65%
Total detection rate	93.08%	98.55%	98.53%	98.47%	98.49%
False alarm rate	0.52%	1.09%	1.11%	1.25%	2.45%
Detection rate for unknown attack	55.46%	54.04%	78.76%	78.92%	76.70%
Number of clustering	14	25	36	47	60

The experimental results show as follows:(1) As r locates in the range of [10,18], the detection results are robust, and the detection rate for unknown attack types is higher than 50%. (2) As $r>20$, the number of clusters increases markedly and time performance descends sharply. (3) As $r<10$, the detection rate goes down and the false alarm rate raises markedly. Considering comprehensively time-efficiency and accuracy, we suggest selecting threshold r in the range of $[EX/3, EX/2]$. (4) Our method outperforms the existing unsupervised methods, so much as supervised method [1], and obtains approving performance. Table 3 shows result in contrast.

The experimental results show that the gravity-based clustering algorithm can produce high quality clusters, and gravity factor can distinguish the normal clusters from attack clusters properly. Table 4 shows contrast labeling results for clusters on P1 by gravity factor and size of cluster, where we let $r=11$ and $\varepsilon=0.05$.

Table 3. The contrast between Ref. [1] and our results

Attack type	DOS	PROBLE	U2R	R2L	All Attack
Ref. [1]	97.1%	83.3%	13.2%	8.4%	91.8%
Our method	99.11%- 99.13%	61.59%- 64.55%	0-17.65%	1.78%- 28.78%	98.49%- 98.55%

Table 4. Labeling results by outlier factor and size of cluster

No.	No. of normal	No. of attack	Labeling by Gravity factor	Labeling by size of cluster
1	0	5	attack	attack
2	2	0	attack	attack
3	11	0	attack	attack
4	0	359	attack	attack
5	248	1136	attack	normal
6	2147	92	normal	normal
7	264	1	normal	attack
8	250	0	normal	attack
9	162	0	normal	attack
10	504	0	normal	attack
11	2198	2	normal	normal
12	26401	3	normal	normal
13	2134	12	normal	normal
14	4506	2	normal	normal

We discover that there are about 30.8%~38.2% clusters only contain two objects or one, most of these objects are ‘normal’ record, all these objects only rate about 0.02%-0.08%. These objects may be aroused by noise and seriously impact detection efficiency. The experimental results show that the detection results change little, and detecting time decrease about 33% if we clean out the noise. So we suggest cleaning out noise to improve the quality of models. All results given in the tables are gained in the case of cleaning noise.

6 Conclusion

In practice, unsupervised detection method is important, because these methods can be applied to raw collected system data and do not need to be manually labeled as an expensive process. In this paper, we introduce the idea of universal gravity to clustering analysis and present a novel-clustering algorithm. At the same time, we present a new unsupervised intrusion detection method, the method needn’t any prior classification about training data and the knowledge about new attacks, and the detection method can detect unknown intrusions. The detection method has the nearly linear time complexity with the size of dataset and the number of attributes, which results in

good scalability. The experimental results show that our method outperforms the existing methods on accuracy.

Acknowledgments

This work is supported by NSFC of P.R. China (No.60273075).

References

1. Charles Elkan. Results of the KDD'99 Classifier Learning Contest. URL: <http://www.cs.ucsd.edu/users/elkan/clresults.html>
2. Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000), 2000, pp255-262
3. Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy and Salvatore Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In Data Mining for Security Applications, Kluwer 2002.
4. Kenji Yamanishi ,Jun-Ichi Takeuchi,Graham Williams,Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In: Proceedings of the Sixth ACM SIGKDD00, Boston, MA, USA, pp 320-324
5. Kenji Yamanishi , Jun-ichi Takeuchi. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner.In:Proceedings of the seventh ACM SIGKDD01,San Francisco, California , 2001,pp389-394
6. Merz C J, Merphy P. UCI repository of machine learning databases. URL: <http://www.ics.uci.edu/mlearn/MLRRepository.html>
7. Portnoy L,Eskin L and Stolfo S J.Intrusion Detection with Unlabeled Data using Clustering.In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001

Anomaly Detection Using Fast SOFM

Jun Zheng, Mingzeng Hu, Binxing Fang, and Hongli Zhang

Computer Network and Information Security Technique Research Center
Harbin Institute of Technology, Harbin 150001, China
{zj, mzh, bxf, zhl}@pact518.hit.edu.cn

Abstract. Different with the host-based anomaly detection, the huge volume of network traffic requires machine learning algorithms more efficient in the network-based anomaly detection. In this paper, the more efficient detection frame based on the SOFM algorithm with the fast nearest-neighbor searching strategy to detect the attack is proposed. We apply the detection frame to DARPA Intrusion Detection Evaluation Dataset. It is shown that the network attacks are detected with relatively low false alarms and more efficiency. The performance of anomaly detection model is improved greatly.

1 Introduction

With the ever fast development of computer networks, the network-based computer security is attracting increasing attention. In addition to intrusion defensive techniques, such as firewall and encryption, Intrusion Detection System (IDS) are used as an important security-barrier against network-based computer intrusions.

There are two general approaches to Intrusion Detection: Misuse Detection and Anomaly Detection. Be similar to virus detection, misuse detection is based on the pattern matching to hunt for the signatures extracted from the known attacks. However, anomaly detection constructing the normal usage behavior profile, named historical or long-term behavior profile. And then anomaly detection analysis model looks for deviations of the short-term behavior profile from the normal usage behavior profiles. And the deviations can be treated as the baselines of estimating the attack activities from normal behaviors.

To date, many machine learning algorithms have been applied to anomaly detection to identify network attacks, including Clustering [1], SVM [2], and Neural Network [3] and so on. However, because of the complexities of algorithms, one of main universal shortcomings of these methods is that these methods are not enough efficient to detect by the real time style. To achieve the on-line detection, we ameliorate the Self-Organizing Feature Map (SOFM) algorithms: the Fast SOFM, adopting the faster nearest-neighbor searching strategy to overcome shortcomings of normal machine learning model. The anomaly detection computing cost is reduced greatly. The performance is very impressive through the comparison.

2 Self-organizing Feature Map

In this paper, the Self-Organizing Feature Map (SOFM) [4] is chosen as anomaly detection model to detect network intrusion activities as deviations from normal profiles exploited by SOFM. The reasons that SOFM is selected are:

- SOFM is one of the unsupervised classification techniques and it is not model-based. We don't need to build the data distribution model. It is important to anomaly detection.
- SOFM is a nonlinear projection of high-dimension data to a lower dimensional space, typically the two-dimension plane. It can be effectively utilized to visualize and explore properties of the data. So by SOFM, we can observe the distributions of the network traffic usage profiles.
- The topology preserving capability and the automatic generation of probabilities for a dataset can make us to explore the relationships among the multivariate traffic flows in the lower dimensional space straightway.

In our anomaly detection, we first use the train data to train SOFM and then trained it to recognize the normal traffic behaviors. By reason of clustering property of SOFM, the process to recognize normal data is to cluster the similar normal traffic behaviors to be substituted with the same node in the SOFM. In detection phrase, the TCP Flows will be input to the SOFM after the form of the feature vectors. By computing the similarities between the input feature vectors and nodes of SOFM, we can distinguish the deviations as attacks from normal profile. The Euclidean distances, named Quantization Errors (QEs), are used to measure the similarities. The process is also known as the nearest-neighbor searching process, searching for Best Matching Unit (BMU).

Because of the character of SOFM, Hoglund [5], Lichodziejewski [6] implemented it as anomaly detection model in the host-based intrusion detection system. All papers used the basic SOFM algorithms without any changes. Because of the complexity of network traffic behavior, it is not difficult to find that the basic SOFM algorithm is too slow to compute when input the large volume of feature vectors in network-based anomaly detection to identify attacks.

2.1 Algorithms Description

First we need to define the TCP Flows as the multidimensional points in Euclidian measurement space:

Definition: Every TCP Flow is a data point in the n-dimension feature space $\mathbf{R}^n$ and $\mathbf{R}^n$ is Euclidian space.

$$\text{TCPFlow} = \{X \mid X \in \mathbf{R}^n\}. \quad (1)$$

Every TCP Flow is expressed by the form of feature vector: $X = (x_1, x_2, \dots, x_n)$.

The following are the main steps involved in SOFM:

The input vector: $X = (x_1, x_2, \dots, x_n)$

The weight vector: $W = (w_1, w_2, \dots, w_n)$

Step1. To initialize every weight vector of SOFM with random values: $w_j(0)$

Step2. To compute the distance between the input vector X_i and the weight vector $W_j(t)$, designate the winner neuron node j^* with the smallest distance. j^* is also called the BMU.

$$j^* = \arg \min_{1 \leq j \leq m} \|X_i - W_j(t)\|. \quad (2)$$

The Euclidean distance is chosen:

$$D = \|X_i - W_j(t)\| = \left(\sum_{k=1}^n (x_{ik} - w_{jk}(t))^2 \right)^{1/2}. \quad (3)$$

D is also called as Quantization Errors (Qes);

Step3. To update the winner vectors of the winner node and its neighborhood:

$$w_{jk}(t+1) = w_{jk}(t) + \alpha(t)[x_{ik} - w_{jk}(t)] \quad j \in N(t). \quad (4)$$

$N(t)$ is the non-increasing neighborhood function, $\alpha(t)$ is learning rate function $0 < \alpha(t) < 1$

Step4. To repeat Step2 and Step3 until SOFM learning stabilizes

2.2 Fast SOFM Algorithm

In anomaly detection, after trained the SOFM, the input feature vectors will be processed to SOFM, which is known as the searching BMU. At the moment Quantization Errors [4] can be got to measure the similarity of short-term behavior and long-term behavior. The SOFM finds the nearest code vector (weight vector) for each input feature vector by exhaustively searching to compute its Euclidean distance to all code vectors in the codebook of SOFM. With the map size of SOFM and input vector increases, the computing cost of Euclidean distances will increase greatly. If the trained SOFM has M vector nodes in the map and the dimension of the flow feature vector is n , to find the BMU, every time SOFM needs to compute n times of power calculation, $(2n-1)$ times addition. So when every TCP Flow feature vector input to SOFM, $M \cdot n$ times of power calculation, $M \cdot (2n-1)$ times of addition and $(M-1)$ times of comparison. In the network anomaly traffic detection, the volume of the inputs challenges the feasibility and the efficiency of anomaly detection. The detection model cannot speed up to satisfy the on-line need.

The basic SOFM is limited in by the computational cost of the full searching. In our methods we take the new the faster nearest-neighbor searching strategy to accelerate computing BMU. This searching strategy was used in the digital image processing fields such as Quantization Vector to get minimum distortion encoding [4].

To suppose input feature vector is $X = (x_1, x_2, \dots, x_n)$; The code vector is $Y = (y_1, y_2, \dots, y_n)$ in SOFM. For X and Y :

$$S_x = \sum_{l=1}^n x_l \quad S_j = \sum_{l=1}^n y_{j,l}. \quad (5)$$

To take the minimum Euclidean distance D is $D(\min)$ in Eq. (3), according to reference [7]:

$$(S_x - S_j)^2 \geq n \cdot D_{\min} . \quad (6)$$

If Y satisfies the Eq. (6), then the Euclidean distance of Y can be avoided and computing can be more efficient. This method is taken between the input vectors and the code vectors in the SOFM. To procure the faster nearest-neighbor algorithm, we will compute the S_j ($j=1,2,\dots,m$) of the code vector Y_j of trained SOFM and sort ascending the Y_j in the value of S_j as shown in Figure 1. In detection phrase, we will search the code vector Y_j which is the nearest neighbor to the input TCP Flow feature vector X by the dimidiate searching algorithm according the sort. Here the weeding rule is the Eq. (6), reducing the compute cost of the Euclidean distance. The dimidiate algorithm is used according to the sort of Y .

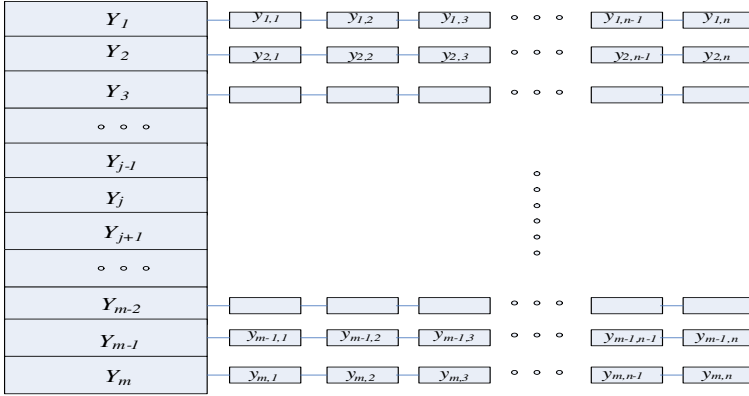


Fig. 1. The Sort of code vectors in SOFM

3 Data Preprocessing

Before anomaly detection process, it is necessary to do Data Preprocessing to extract the feature attributes from IP packets, and then, the Date Normalization will be processed to project whole feature attributes to a unit range. In the paper, data preprocessing is focused on TCP traffic. Because of complexity and vulnerability, TCP acts as two roles mainly: network attack carrier and network attack target. In the IP traffic of Internet, TCP accounts for 95% or more of the bytes, 85-95% of the packets [8]. Moreover, according to the statistical data from Moore [9], the majority of DDoS/DoS attack which is main threat to the whole Internet is deployed by using TCP as 90~94%. So the paper is focused on the TCP traffic merely and constructs the light anomaly detection system.

The extraction of feature attributes of network traffic is the foundation of machine learning algorithms in anomaly detection. Moreover, excellent detection models or algorithms must be combined with the rational feature vector extraction to improve the attack recognition capability. Traffic features should prefer to differentiate usual traffic

profiles from anomaly traffic profiles. The aim of feature extraction is to achieve the maximum difference degree between usual usage behaviors and anomaly behaviors. A feature vector of the traffic flow is shown in Table 1.

Table 1. Feature attributes of TCP Flow feature vector

Feature Attribute	Describe
SrcIP	source IP address
DestIP	destination IP address
SrcPort	source port
DestPort	destination port
PktSize	average packet size in one TCP Flow
SrcBytes	the number of bytes from source
DestBytes	the number of bytes from destination
FlowState	TCP Flow closed state
Fre_SrcIP	frequency of a certain source IP in time-window
Fre_DestIP	frequency of a certain destination IP in time-window

4 Evaluation Method and Result

4.1 Experiment Data Set

We take a part of the Intrusion Detection Evaluation Data Set of 1999 DARPA [10] to estimate our anomaly detection off line. SOFM is trained on the data set (inside traffic only) attack free in week 3 and week 1. However, consider the fact that the target of this paper is network work attack based on TCP and not general, we test for TCP attacks (DoS and Scanning) merely. The original test data seem colossal to us so the inside traffic test data of week 4 and week 5 is condensed to 23623 items of TCP flows of the original test data. We filter out some other attacks out of the test traffic data according to the attack identification [11] of 1999 DARPA after the feature vectors extracted. Five types of attack, 22 instances total, are used to test as in Table 2. A detailed description of these attacks could be found in [11].

Table 2. Attacks in test

Attack Name	Mailbomb	Portseep	Queso	Resetscan	Tcpreset
Number	3	11	4	1	3

4.2 Evaluation Method and Result

After training, traffic TCP flows of test are input to the SOFM in term of the feature vector. At last, the Map is build to contain the training data. The train data is detached by day to train the normal usage SOFM. We use three different SOFMs, with the map sizes of 40×40, 30×30 and 25×25 to test the difference of detection time between the basic SOFM and the fast SOFM with nearest-neighbor searching strategy. The comparison outcome is very impressive in fact of computing time cost during the detection

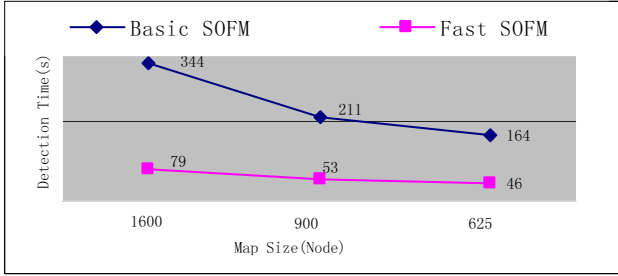


Fig. 2. Time cost for different map size

phrase as presented in Figure 2. The total 23623 items of TCP flows are used in the test. The experiment computer is Dawning Sever (Linux7.2/PIII800/RAM1G).

From the Figure 3, we can see that Quantization Errors (QEs) of attack samples are very prominent. By the modulation of QE threshold with 900 map nodes, we can get the satisfactory detection rate with low false alarms displayed in Table 3 (threshold $QE=5.0$). Because of attack duration, the number of QE spikes displayed in Figure 3 is not equal to the real number of attack instances, 22.

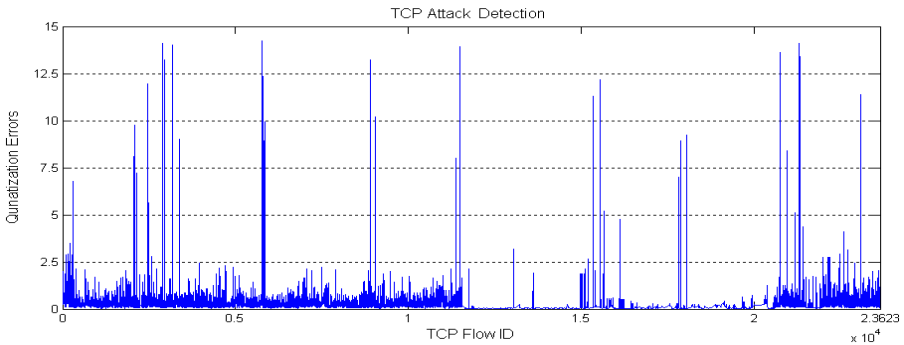


Fig. 3. Quantization errors

5 Conclusion

The problem of efficiency is the one of main obstacles that blocked machine learning algorithms to be applied in anomaly detection in the network security fields. In this paper, we proposed the fast SOFM with the nearest-neighbor searching strategy in anomaly detection, especially for detecting some types of TCP attack. The new algorithm reduced the whole detection time cost and enhanced the capacity of real-time intrusion detection. Moreover, we took the TCP flow as the basic data unit in data preprocessing. The evaluation experiments confirmed that the fast SOFM can achieve the higher detection rate with the lower false detection rate.

Table 3. Test result

Flow ID	QE	Attack
288	5.438	portsweep
312	6.761	portsweep
2067	8.087	portsweep
2084	9.739	portsweep
2135	7.223	portsweep
2465	11.938	mailbomb
2471	10.987	mailbomb
2893	14.122	mailbomb
2945	13.214	mailbomb
3183	14.012	×
3371	8.992	portsweep
5754	14.213	×
5774	11.122	mailbomb
5789	12.329	mailbomb
5791	10.902	portsweep
5809	8.923	portsweep
5837	9.913	tcpreset
8882	12.413	queso
8894	13.245	queso
9048	10.213	×
11378	8.011	tcpreset
11485	13.921	queso
15334	11.312	queso
15534	12.193	portsweep
15535	6.349	portsweep
15658	5.176	×
17806	6.991	resetscan
17854	8.932	resetscan
18022	9.211	portsweep
20737	13.626	portsweep
20943	8.382	tcpreset
21165	5.101	portsweep
21283	14.112	queso
21299	13.391	queso
23072	11.381	×

(× means false detection with the threshold QE=5.0)

Acknowledgments

The work was supported by the National High Technology Development Program of China 863 (Grant No. 2002AA142020) and (Grant No. 2001AA147010B)

References

1. E. Eskin, A. Arnold, M. Prerau: A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. *Applications of Data Mining in Computer Security*, Kluwer, 2002
2. Andrew H. Sung, Srinivas Mukkamala: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. Proceedings of the 2003 Symposium on Applications and the Internet (2003)
3. Srinivas Mukkamala, Andrew H. Sung: Identifying key features for intrusion detection using neural networks. Proceedings of the 15th international conference on Computer communication. Aug 2002.
4. Kohonen, T.: Self-Organization and Associative Memory, 3rd ed., Springer-Verlag, Berlin, Germany 1989.
5. A. J. Hoglund, K. Hatonen: A computer host-based user anomaly detection system using self-organizing map. IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5, 2000
6. P. Lichodziejewski, A. N. Zineir-Heywood, M. I. Heywood: Host-based intrusion detection using self-organizing Maps. Proceedings of the 2002 IEEE IJCNN, Hawaii, USA, May 2002.
7. T. Torres, J. Huguët: An improvement on codebook search for vector quantization. IEEE Trans. Commun. , 42 (2) :2080-210, 1994
8. G. J. Miller K. Thompson and R. Wilder: Wide-area Internet traffic patterns and characteristics. IEEE Network, pages 10-23, Nov 1997.
9. D. Moore, G. Voelker, and S. Savage: Inferring Internet Denial-of-Service Activity, in Usenix Security Symposium, Washington, D.C., Aug 2001
10. Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, Kumar Das: The 1999 DARPA Off-Line Intrusion Detection Evaluation, Computer Networks, 34 (4): 579-595, 2000
11. 1999 DARPA Intrusion Detection Evaluation Dataset.
URL: <http://www.ll.mit.edu/IST/ideval/index.html>

Research on Secure Multicast Technology in Grid-Based Large-Scale Distributed Simulation Applications*

Wei Wu, Ling Shen, Haitao Huo, and Xiaojian Li

The Key Laboratory of Virtual Reality Technologies, Ministry of Education
BeiHang University, Beijing 100083, P.R. China
{wuwei, shenling, hht}@vrlab.buaa.edu.cn

Abstract. Security problems in application layer multicast communication oriented to grid-based large scale military simulation applications are discussed. On the basis of grid services and security architecture, the mechanism of grid secure multicast services (GSMS) is proposed. GSMS provide functions of access control and group key management to ensure the confidentiality, integrity and non-repudiation of the multicast information in the multicast communication, and also enable the abilities to examine the system status, inspect abnormality, log the system, enhance system security and improve the capability to resist attack. Finally, an application instance is used to validate the ability of GSMS in Globus Toolkit 3.2 platform, indicating the usability of GSMS.

1 Introduction

Multicast is widely used in large-scale distributed simulation applications in order to reduce the consumption of network bandwidth. As grid technology grows, more and more simulation applications are constructed on grid technology platform [1]. In grid architecture [2], grid security information infrastructure provides a set of security protocols and services to sustain security communication between the entities in grid environment. However, these services are mostly oriented to unicast communication, and cannot provide special services for multicast communication.

Current multicast protocols provide neither user authentication support nor credible security guarantee. Every user can arbitrarily join a multicast group, send multicast messages, and leave without announcement. Multicast sender can not know the time when receivers join or leave, and can not calculate the number of multicast receivers. Therefore, although multicast technology has advantages to develop new operations, there are still lots of security problems. Paper [3] points out that secure multicast must solve four problems: multicast data confidentiality, multicast group key management, multicast data source authentication, and multicast security policies.

Internationally, a lot of research work on multicast security has been made. For example, GKMP[4], Clique[5], Iolus[6], and OFT[7] have been put forward for multicast key management, aiming at solving distribution and updating problem of group key in large scale of multicast application. There are also many multicast data source

* This paper is supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312105.

authentication solutions such as TESLA and other solutions which base on MAC and digital signature[8]. Generally speaking, every scheme has its own advantages and disadvantages[3], and we must choose proper solution depending on characteristics of given multicast application. This paper will focus on the resolution of multicast security problem in distributed simulation applications in the grid environment.

2 Model of Secure Multicast Communication System Based on Grid Services

This paper designs secure multicast communication services (GSMS) to resolve the security problems of multicast communication in grid environment.

2.1 Grid Secure multicast Communication Services

Grid Secure Multicast Services (GSMS) is used to protect the confidentiality, integrity and irreversibility of multicast content, authenticate data source, and resolve controlled login and logout of the group members. GSMS include five services:

1. Group Authorization Service(GAS): it provides access control of the group members and the management of the group key in multicast communication for the service users who usually are the group managers of the multicast communication.
2. Security Transfer Service(STS): it provides encryption and decryption of the multicast communication data for the service users who usually are the group members of the multicast communication, encrypting the sending multicast information and decrypting the received multicast data.
3. Source Authenticate Service(SAS): it makes signature or checks the source of the multicast data so that the group member can ensure the data source and the data sender can not deny the data he sent out.
4. Group Monitor & Control Service(GMCS): it provides the system status and the statistics of the system condition for the service user on basis of information analysis of the multicast system communication. Furthermore, GMCS offers automatic control service to make response to the abnormal situation in multicast communication as well as carry on defense to possible attack according to certain rules.
5. Group Log Service(GLS): it provides system logs for the service users.

Among above five services, GSA, STS and SAS are three basic services of GSMS, synthetically application of which can secure the confidentiality, integrity and non-repudiation of the multicast information in the multicast communication. GMCS and GLS are two expanded services, which can help to examine the system status, detect abnormality and record system logs to enhance the capability of usage and audit and improve the ability to resist attack.

2.2 Architecture of GSMS-Based Secure Multicast Communication System

The hierarchy architecture that can be adopted in practice has shown in Fig.1. The Group Members (GMs) are divided into several subgroups, and every subgroup is managed by a Group Authorization Agent (GAA). All the GAAs compose a higher

level group which is managed by Group Authorization Manager Center (GAMC). The sub groups use the same group key that is generated and refreshed by the GAMC. The group key is refreshed at regular intervals to adapt to the large-scale distributed application, that means the group key is not refreshed when group member joins in or exit the multicast group.

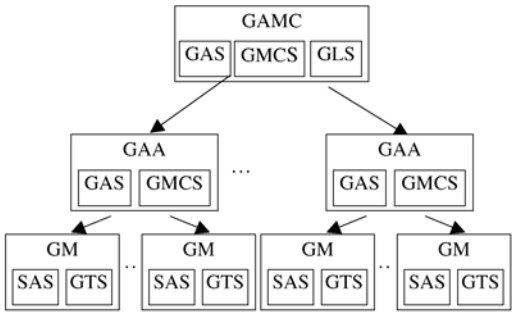


Fig. 1. System architecture using GSMS

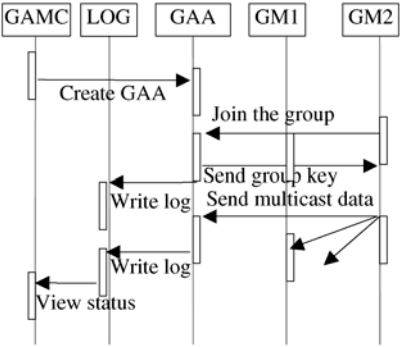


Fig. 2. Multicast system scheduling

The scheduling of the system is shown in Fig.2. After group administrator starts up the GAMC, the entire group system can be managed through GAS, GMCS and GLS. GAAs created by GAMC and assist the GAMC to manager group by using GAS and GMCS. GMs join in the multicast group, send and receive the multicast data securely through SAS and GTS.

3 Implementation of GSMS

The implementation of GSMS is constructed on Globus Toolkit [10] platform. Based on the Grid Security Infrastructure (GSI) [9] and other grid services such as Global Access to Secondary Storage (GASS), GridFTP etc., and GSMS can make full use of existing grid resource to provide a security assurance for grid application.

3.1 Group Authorization Service

Group Authorization Service(GAS) can provide an interface to control GM and manage the group key of the multicast communication. Simultaneously, it can create GAAs dynamically to manage multicast group collaboratively according to the scale of multicast group. After GAS starts up, it first initializes, and then deal with two types of service request and one timer message. The workflow is shown as the Fig. 3.

1. Initialization

GAS starts up GMCS according to the initial information, then starts up the GAS of GAAs. The detailed algorithm for the generation of one GAA is described below.

- (1) Obtain the usage right of the network node which will be the GAA.
- (2) Generate a temp security certificate Cag' for the GAA.
- (3) Sign Cag' and the certificate which has been signed are shown as below.

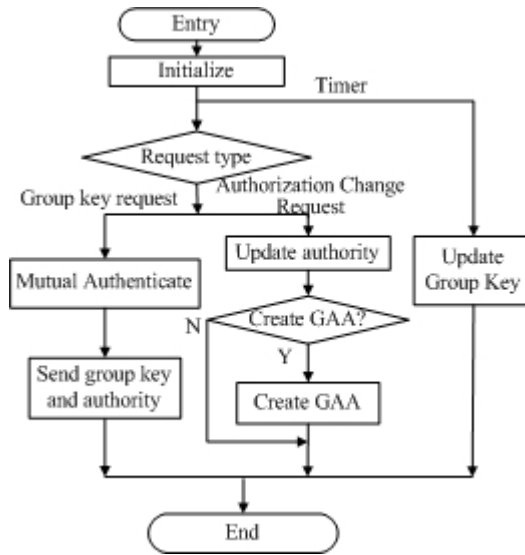


Fig. 3. Flow chart of GAS

$$\text{Cag} = \text{Cm}\{\text{Cag}', \text{start-time}, \text{end-time}\} \quad (1)$$

Cag includes the temp security certificate and the survival time of the certificate and Cm is the certificate of administrator.

(4) Generate GAA, and provide security identity assurance through the signature of the security certification.

Those processes will be accomplished by the services provided by GSI.

2. Group key request

Multicast user sends group key request when joining multicast group. GAS accepts the request, validates user's certification first, then sends the group key and his authority information to the multicast user, according to the authorization of the system (such as multicast information sending right, receiving right, etc.).

If system starts up the GMCS, GAS will also send group member's authorization and authentication information, including group member's name, permission right, joining time, IP address and multicast address which group member want to join in.

3. Authorization change request

Multicast group administrator sends this request when he changes the group Authorization or finds some abnormal condition. GAS will also decide whether to create a new agent or not according to the number of new group members.

4. Timer

Timer is used to update the group key at regular intervals. GAMC creates a new group key and sends the notification to the group members. GAA receives the group key to make sure the GAMC works normally. If GAAs fails to receive the new key, they will elect a temp GAMC according to their survival time. Because GAMC will set the valid survival time when creating the agent, GAAs will select the agent whose certification will be invalidated last to maintain the multicast communication.

3.2 Security Transfer Service

Security Transfer Service (STS) encrypts or decrypts the multicast communication information, and maintains the group key it used. STS deals with two types of service requests: Information transfer request and Group key update notification.

1. Information transfer request

Multicast user sends an information transfer request when he wants to send or receive the multicast information. STS first authenticates with GAS to acquire the group key and the user's authorization after accepting the user's request, then encrypts the sending information or decrypts the received multicast data using the group key based on the user's authorization, and uses SAS to sign the sending information or encrypt the signature of the received multicast data to confirm the data source when needed.

2. Group key update notification

GAMC sends group key update notification at regular intervals, GTS authenticates with the GAMC to acquire the new group key and the user's authorization when receiving the notification.

3.3 Source Authenticate Service

Source Authenticate Service (SAS) signs or encrypts the signatures of the multicast data. SAS adds timestamp and other information to resist playback attack when signing the sending information. As for the received multicast information, SAS queries the certification of the sender who is claimed in the multicast data and distills his public key from the certification to encrypt the signature of the received multicast data.

3.4 Group Monitor & Control Service

Group Monitor & Control Service (GMCS) shows the run-time status of the multicast system. It calculates current multicast groups and the numbers of group members in every group based on the authentication and authorization information received from GAS. Simultaneously, GMCS will join every multicast group as a special group member, receive the multicast data from the group members, confirm the data source, decrypt the multicast data and record all those information in the log. The workflow is shown in the Fig. 4.

Moreover, GMCS can find the abnormal situation (such as decrypting the multicast data unsuccessfully, group member sending multicast data beyond his authorization, etc) based on the analysis of the authentication and multicast communication information, produce and send corresponding action (such as notifying the group member to authenticate again, changing the authorization of group member, expelling a group member from his group, etc) to GAS.

3.5 Group Log Service

Group Log Service (GLS) accesses the log files which are distributed in different locations through GASS and GridFTP which are provided by Globus to offer the view of three kinds of system log.

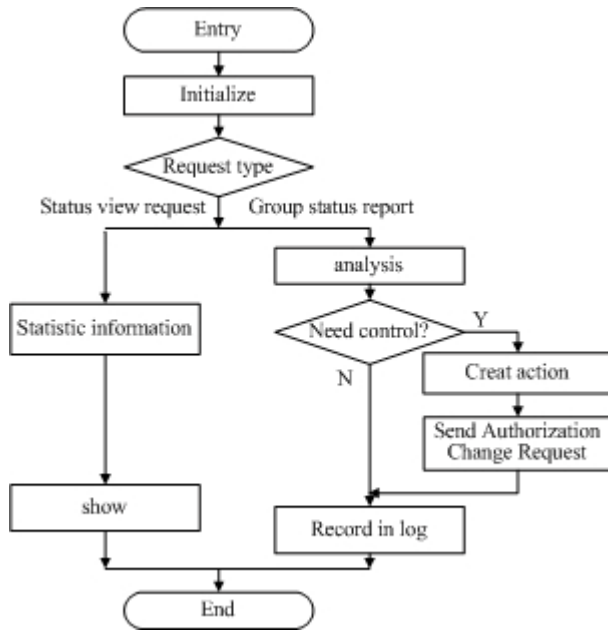


Fig. 4. Flow chart of GMCS

1. User authentication and authority log: records the users' name, logging time, IP address, multicast address and corresponding authorization.
2. Multicast data log: records the sender name, destination address and sending time of the multicast data and the sender's IP address.
3. Abnormal response log: records the type of the abnormal status, the action manner, response time, related user name and multicast address.

4 Instance Based on GSMS

In large scale distributed simulation applications, time synchronization is a basic requirement. In the following, this paper uses a multicast time synchronization program to validate the usability of GSMS. This paper use Globus Toolkit 3.2 as grid platform, and implement this instance under Sun Solaris 8 operation system. Fig. 5 shows the environment in the instance.

The program running process is described below:

1. The administrator starts up GAMC program, and then uses GAS to authorize. For example, user A, B, C and D want to join multicast group 234.5.6.7 and communicate with each other, The administrator assigns encryption algorithm type (we now implement three symmetry encryption types: DES, IDEA, RC5), key, initialization vector and group permission(send, receive, send & receive).
2. Start up time synchronization server and client programs as group members to send and receive multicast information. Time synchronization server program uses STS to encrypt data and then use SAS to make signature. Client program uses STS to

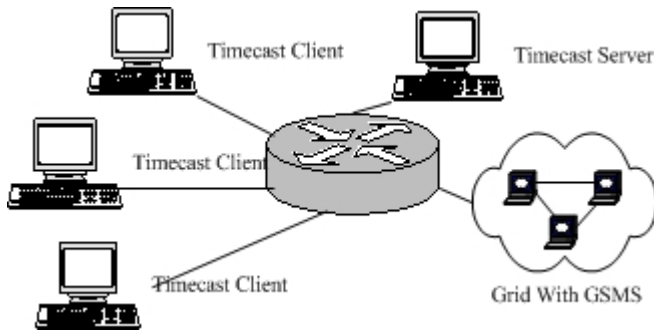


Fig. 5. Time synchronization environment

decrypt data and then use SAS to confirm multicast data source. Server will send one time synchronization value every second, and only group users with receiving permission can read information correctly. At the same time GMCS can record the logs.

3. The administrator uses GMCS to examine current multicast runtime status, logs and so on.

5 Conclusion

This paper designs a security multicast communication system based on grid security multicast services to resolve the multicast security problems in distributed simulation. Grid security multicast services provide five services: GAS, STS, SAS, GMCS and GLS. Among those five services, GSA, STS and SAS are the three basic services of GSMS and responsible for access control to group members and encryption and sign of multicast information in multicast communication so as to ensure the confidentiality, integrity and non-repudiation of the multicast information in the multicast communication. GMCS and GLS are two expanded services and help to examine the system status, find out abnormality and record system logs to enhance the capability of use and audit and improve the ability to resist attack. This paper adopts the hierarchy system architecture to implement the security multicast communication based on grid security multicast services in grid environment. The experiment indicates the usability of GSMS.

References

1. K. Zając, A. Tirado-Ramos, Z. Zhao, P. Sloot, M. Bubak. Grid Services for HLA-based Distributed Simulation Frameworks. In European Across Grids Conference, 2003.
2. Ian Foster, Carl Kesselman, ed. The Grid 2: Blueprint for a New Computing Infrastructure, 2nd edition, Morgan Kaufmann, Nov. 2003.
3. Hardjono T and Tsudik G (2000) IP Multicast Security: Issues and Directions. Annales de Telecom, July-August 2000, pp. 324-334.
4. Harney H, Muckenhirn C. Group key management protocol (GKMP) architecture. RFC2094, 1997.

5. Setiner M, Taudik G, Waidnet M. Cliques: A new approach to group key agreement. Technical Report, RZ 2984, IBM Research, 1997.
6. Mitra S. Iolus: A framework for scalable secure multicasting. In: ACM SIGCOMM Computer Communication Review, Volume 27, Issue 4. New York: ACM Press, 1997. 277~288.
7. Balenson D, McGrew D, Sherman A. Key management for large dynamic groups: One-Way function trees and amortized initialization. IETF Internet Draft (work in progress), 2000.
8. Wong CK and Lam SS (1998) Digital Signatures for Flows and Multicasts. Proceedings of IEEE ICNP '98, Austin, TX, October 1998.
9. The Global Grid Forum, www.gridforum.org.
10. The Globus Alliance, www.globus.org

BSCM: Proposed Security Collaboration Model Based on Blackboard

Yong Tang, Hua-Ping Hu, and Xicheng Lu

School of Computer Science, National University of Defense Technology
Changsha Hunan, 410073, P.R. China
tanyoo10@163.com

Abstract. Security collaboration is necessary to improve integral security of network. But there is no unified model to ensure interoperability and collaboration within IDS, firewall and other network security components. We propose a security collaboration model by exploiting blackboard technique, named BSCM. In this model, network security components don't communicate with each other directly, but via a common blackboard which serves as the platform of information-sharing and events-responding. We introduce the "Register-feedback" mechanism which connects the blackboard and all the components in BSCM. The formal definition of BSCM and a sample system are given in this paper.

Keywords: Network security, Security collaboration model, Blackboard

1 Introduction

Network security has become an increasing problem in the field of information technology. To maintain network's security, currently used computer security systems usually consist of a number of network security components, includes: firewall, intrusion detection system, vulnerability scanner, anti-virus etc. These distributed security components usually require an enormous amount of distributed and specialized knowledge to approach their security objective. However, the lack of collaboration within them has restrained the improvement of integral network security.

There have been many studies on the exploitation of collaborations within network security components. Several researchers have studied applying the cooperative distributed agents to information security system [1], especially to network intrusion detection [2, 3, 4]. Carver proposed a methodology for adaptive, automated intrusion response (IR) using software agents [5]. Schnackenberg presented Cooperative Intrusion Traceback and Response Architecture (CITRA) as an infrastructure for integrating network-based intrusion detection systems, firewalls, and routers to trace attacks back to their true source and block the attacks close to that source [6]. OPSEC (Open Platform for Security) [7] is the industry's open, multi-vendor security framework for interoperability.

However, little research has been devoted to unified security collaboration model, which can ensure interoperability and collaborations within arbitrary security components. A unified security collaboration mode would have the following features:

- Provide framework to integrate all kinds of security components
- Provide mechanism for communication and cooperation
- Help to exploit certain security technique such as distributed intrusion detection
- Support integrated and centrally security management and improve integral network security

We propose a security collaboration model; named BSCM (Blackboard based Security Collaboration Model). In this paper, the framework of BSCM and BSCM's "register-feedback" mechanism is presented first. Then we give the formal definition of BSCM. Finally, an example system according to BSCM is discussed.

2 Blackboard Technique to Build BSCM

The Blackboard Technique is a popular AI technique used for problem solving. A blackboard system can be viewed as a collection of intelligent agents (called knowledge sources) who are gathered around a blackboard, looking at pieces of information written on it, thinking about the current state of the solution, and writing their conclusions on the blackboard as they generate them [8]. A blackboard system consists of a set of knowledge sources, a blackboard data structure, and a control strategy used to activate the knowledge sources. The blackboard is a centralized global data structure, often partitioned in a hierarchical manner, used to represent the problem domain. The blackboard is also used to allow inter-knowledge source communication and acts as a shared memory visible to all of the knowledge sources [9].

The blackboard technique has been employed in various applications and real-time systems. Some of them are ATOME [10], GBB [11]. The blackboard techniques has also been used in the network security domain. Dasgupta described how blackboard-based agent architecture helps in detecting intrusions [12]. Dass discussed the design of a Learning Intrusion Detection System (LIDS) that includes a blackboard-based architecture with autonomous agents [4].

The blackboard architecture is considered as one of the most general and flexible knowledge system architectures for building decision-based applications. It is highly preferred over other alternatives due to its modularity, dynamic control, concurrency, and ability in dealing with multiple knowledge sources.

Therefore, we use the blackboard technique to build BSCM. The blackboard architecture is used as the framework to integrate security components. Then we propose a "Register-feedback" mechanism. By this mechanism, network security components which need collaborations must communicate with the blackboard.

3 BSCM: Proposed Security Collaboration Model

3.1 Framework

BSCM uses blackboard architecture as the framework to integrate all kinds of security components. Figure1 provides an overview of the BSCM framework. The system consists of a *blackboard*, a *controller*, a *GUI*, a collection of *network security components*.

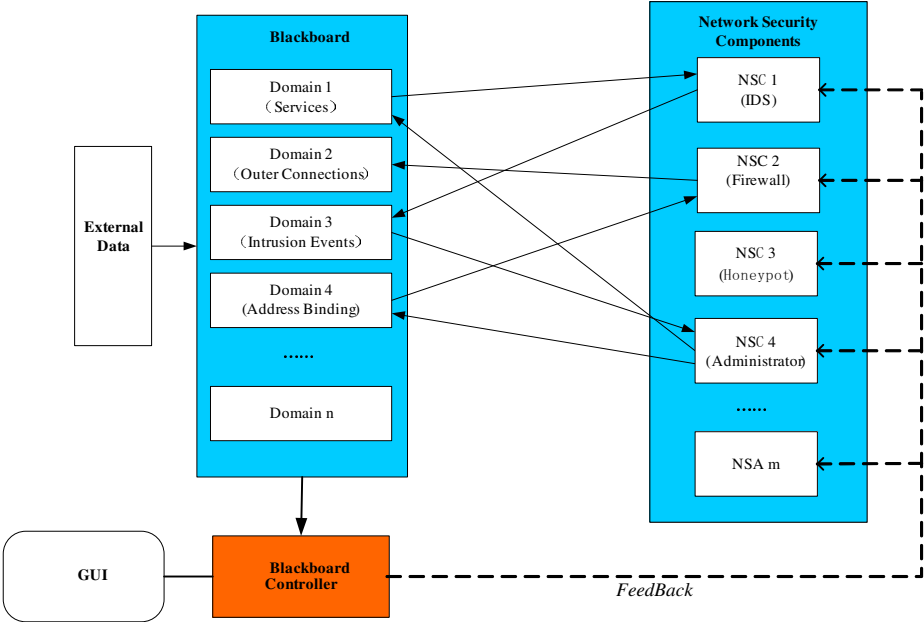


Fig. 1. Framework of BSCM

The blackboard is the centralized data unit of the system. The valuable information about network or network security is placed on the Blackboard. The real-time events created by network security components are also placed on the Blackboard. All of information is divided into many groups according to usage; each group is called a domain. For example, a domain can be connections from outside, trustworthy hosts, suspicious hosts, intrusion alerts etc. A domain has some fields and contains records like a table in database. Moreover domains can be added, removed, or updated dynamically.

Network security components are all kinds of security products, tools, or components which are parts of an integrated network security system. Especially, the administrator of a network can also be seen as a network security component. They can write configuration information of networks or hosts, security policies, experiential knowledge and other valuable information into the blackboard. As a result, network security components can collaborate not only with other components but also with humans. And the knowledge and experiences of humans would help some network security components improve their usability or performance.

The blackboard controller is mainly used to monitor and control the blackboard, and implement the “register-feedback” mechanism (See details of “register-feedback” mechanism at the sector 3.2). Because the controller also acts as the medium, through which network security components access the blackboard. The controller must assure the security demands such as communication encryption and identity verification.

The Graphic User Interface (GUI) not only can display the running status of the blackboard, but also can give the whole view of the integral security system.

3.2 “Register-Feedback” Mechanism

In BSCM, the network security components don't communicate with each other directly. Instead they communicate and cooperate via the common blackboard. Firstly the network security components register some domains, which they are interested in. Once granted by blackboard controller, network security components could access these domains whenever they need, or duplicate these domains to local considering the efficiency. When the contents of a domain have been modified, the blackboard controller will notice the network security components which registered this domain. We call this reaction between the blackboard and the network security components “Register-feedback” mechanism. The collaborations of BSCM are achieved by this “Register-feedback” mechanism in fact.

Under the “Register-feedback” mechanism, the network security components work in real-time to take special security responsibility. Their inputs can be obtained from the environment of network, and also the domains they registered. Their result information can be exported to the blackboard, and then can be read by other network security components. Further, when a network security component triggers and records an event in blackboard, some other network security components could respond for this event.

The “Register-feedback” mechanism is simple but makes possible that network security components needn't know the positions of other network security components, which they want to collaborate with. All the information they need can be obtained from the blackboard. A benefit of this mechanism is that all kinds of network security components need only implement just one interface for communicating with the blackboard. Another advantage that this mechanism offers is the extensibility in architecture; new network security components can be adding to the integrated security system under BSCM.

3.3 Formal Definition

The formal definition of BSCM is presented as following:

BSCM = $\langle \Omega, \Phi, \mathbf{B}, \text{Role}, \text{Responsibility}, \text{Capacity}, \text{Register}, \text{Feedback} \rangle$, where Ω is the network environment, including everything that could reflect the network's status, for example, the packets of networks and the activities of users.
 Φ is the set of network security components.

$$\Phi = \{ A_1, A_2, \dots, A_n \}$$

B is the blackboard, which is a set of *domain*. A *domain* is a set of information or events which describing the network environment. The contents and the format of *domains* can be different, that offers the flexibility.

$$\mathbf{B} = \{ D_1, D_2, \dots, D_m \}$$

Role is the set of security functions, which act roles in maintaining the security of network. In a network security system, these functions include access control, intrusion detection, traffic monitor etc.

$$\text{Role} = \{ rol_1, rol_2, \dots, rol_k \}, \text{ where}$$

$$rol_i = \Omega \times \bigtimes_{D_j \in B} D_j \rightarrow D_i \quad (D_i \in \mathbf{B}, 1 \leq i \leq k). \quad rol_i \text{ refer to a security function.}$$

It means that a network security component reads the information from the environment and some *Domains* in the blackboard, then executes special action, and outputs result to certain *Domain*.

Responsibility is a dispatch of roles in **Role**. Let A^{rol_i} ($A \in \Phi, rol_i \in \text{Role}$) represents the network security component A can fulfill a function defined in rol_i , then a network security components' all capabilities in the network security system can be defined as a set:

$$\mathbf{Capacity}(A) = \{ A^{rol_{i1}}, A^{rol_{i2}}, \dots, A^{rol_{ir}} \}. \text{ So,}$$

$$\mathbf{Responsibility} = \sum_{i=1}^n \mathbf{Capacity}(A_i) = \{ \{ A_1^{rol_{s1}}, \dots, A_1^{rol_{sp}} \}, \dots, \{ A_n^{rol_{t1}}, \dots, A_n^{rol_{tq}} \} \} \quad (1 \leq s_1, s_p, t_1, t_q \leq k)$$

Register represents the Register mechanism of the blackboard.

$$\mathbf{Register} = \{ \langle A_{i1}, D_{i1} \rangle, \langle A_{i2}, D_{i2} \rangle, \dots, \langle A_{ik}, D_{ik} \rangle \} \quad (\langle A_{ik}, D_{ik} \rangle \in \Phi \times \mathbf{B}),$$

Where, $\langle A_{ik}, D_{ik} \rangle$ represents network security component A_{ik} registered in domain D_{ik} .

Feedback represents the reaction mechanism of the blackboard. Blackboard follows the rule below, called **Feedback**:

If Changed(D_i)

then for $\forall A_j$, if $\langle A_j, D_i \rangle \in \mathbf{Register}$, then Notice(A_j, D_i).

Changed(D_i) ($1 \leq i \leq m$) means that the domain D_i has been modified or updated,

Notice(A_j, D_i) means that the blackboard notice network security component A_j that domain D_i has changed.

So, the "Register-feedback" mechanism of BSCM is defined by the **Register** and **Feedback**.

4 An Example System Based on BSCM

Consider a simple but integrated network security system based on BSCM as following.

4.1 Constitution of the System

There are three network security components (IDS, firewall and administrator) and six domains in the system. According to the definition of BSCM, we obtain

$$\Phi = \{\text{IDS, Firewall, Administrator}\}$$

$$\mathbf{B} = \{\text{D1, D2, D3, D4, D5, D6}\}.$$

The explanations of the domains are given in Table 1. Table 2 describes the relations between the network security components and the blackboard.

Table 1. Domains in the example system

Domain	Meaning	Fields
D1	Description of local network service	IP, ServiceName (the supporting service), Port(the port of service),Platform (operation system), IsActive (active or not)
D2	Trusty hosts	IP, Platform
D3	Suspicious hosts	IP, Reliability, AttackTimes (times of suspicious action or attack)
D4	Alerts from IDS	SIP(source address), DIP(destination address), Time, Severity(Severity of the event), Reliability(possibility of false alert)
D5	Address Binding	IP, MAC
D6	Access Control Rules for the Firewall	Direction, SIP, DIP, Service, Time, Action (accept, block, or reject) ,Log (log or not)

Table 2. Relations between the network security components and the blackboard

Network security component	Registered domain	Operational domain	Explanation
IDS	D1, D2, D5, D6	D3, D4	Monitoring the network traffic in real-time, and considering information in the domain D1, D2, D5, D6, IDS detects attacks and suspicious actions. The suspicious hosts are recorded in D3; the alerts are recorded in D4.
Firewall	D3, D5, D6	D6	Firewall autonomously adjusts its security policy according to the domain D3, D5, D6.
Administrator	D3, D4, D6	D1,D2, D5,D6	Administrator sets the network environment variables (D1, D2, D5), the security police(D6), according D3, D4, D6.

4.2 Collaborations Within Network Security Components

Figure 2 shows the collaborations within the network security components in the example system. Directions refer to collaborations, as shown below.

1. If the IDS detect attacks or suspicious actions, the domain of D3, D4 will be updated, and then the administrator who registered these domains will known in time.
2. The environmental variables that D1, D2, D5, which are set by administrator, are valuable information for IDS to improve the detect precision and reduce the false

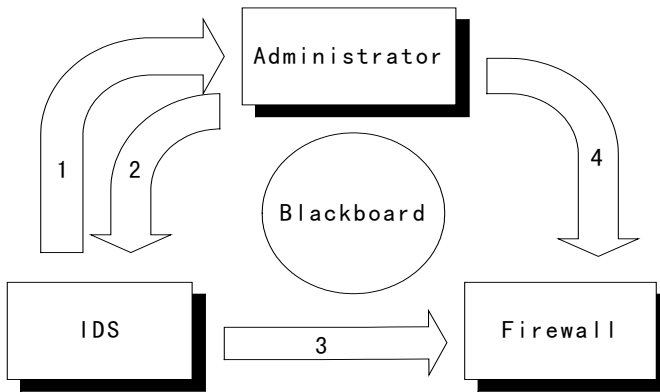


Fig. 2. Collaborations within the example system based on BSCM

alerts. This case shows the flexibility of the BSCM that human can help network security components' work by providing knowledge.

3. In case the IDS detects alerts or suspicious actions from outside, the firewall will decide whether to modify the access control rules in order to block the attack or not, according the Severity fields and the Reliability fields in D4. In this way, security events can be responded by other network security components.
4. If the administrator modifies the domain D5 or D6, the firewall will adjust its local attributes and change running status accordingly.

5 Conclusions

BSCM provide a way to integrate all kinds of network security components, thereby improving the integral security of network. Network security components share information and respond for some events by the way of "Register-feedback" mechanism in BSCM. An advantage of BSCM is that all network security components collaborate by implementing only one communication interface to blackboard. Another advantage of BSCM is the extensibility and flexibility that the knowledge and experiences of human can be brought to integrated security system. Currently, we are implementing a prototype for our model according to the example system described in this paper.

References

1. V.I. Gorodetski, L.J. Popyack, I.V. Kotenko: Ontology-based Multi-agent Model of Information Security System. Lecture Notes in Artificial Intelligence, No. 1711, 1999.
2. Balasubramanian, J., Fernandez, J. O., Isacoff, D., Spafford, E., and Zamboni, D.: An Architecture for Intrusion Detection Using Autonomous Agents. In COAST Technical Report 98/5, Purdue University, 1998.
3. M.Asaka et al.: A Method of Tracing Intruders by Use of Mobile Agents, in Proc. of INET'99, June 1999.

4. M.Dass, J.Cannady, W.D.Potter, LIDS: Learning Intrusion Detection System, American Association for Artificial Intelligence. [http:// www.aaai.org](http://www.aaai.org), 2003.
5. Carver, C. A., Hill, J. M. D., Sudru, J. R., and Pooch, U. W. 2000. A Methodology for Using Intelligent Agents to Provide Automated Intrusion Response. In IEEE Systems, Man and Cybernetics Information Assurance and Security Workshop, West Point, N.Y., June 2000.
6. D.Schnackenberg, K.Djahandari, and D.Strene, Harley Holiday, Randall Smith: Cooperative Intrusion Traceback and Response Architecture (CITRA), in Proceedings of the 2nd DARPA Information Survivability Conference and Exposition (DISCEXII), June 2001.
7. OPSEC (Open Platform for Security), <http://www.opsec.com/>
8. Nii, H. Penny: "Blackboard Systems: The Blackboard Model of Problem-solving and the Evolution of Blackboard Architectures." *AI Magazine*, 7 (2), pp. 38 - 53, 1986.
9. McManus, John W.Design and Analysis Tools for Concurrent Blackboard Systems. In Proceedings 10th AIAA/IEEE Digital Avionics Systems Conference, October, 1991.
10. Laasri, H., B. Maitre, T. Mondot, F. Charpillat, and J-P. Hatton.: ATOME: Another Tool for Developing Multi-Expert Systems, Proceedings of AAAI-Workshop on Blackboard Systems, Seattle, Washington, pp.309-322, 1987.
11. Corkill, D., K. Gallagher, and K. Murray.: GBB: A Generic Blackboard Development System, in Blackboard Systems, edited by R. Englemore and T. Morgan, Addison-Wesley,1988.
12. Balasubramanian, J., Fernandez, J. O., Isacoff, D., Spafford, E., and Zamboni, D.: An Architecture for Intrusion Detection Using Autonomous Agents. In COAST Technical Report 98/5, Purdue University, 1998.

Credential Trustworthiness-Based Trust Evaluation in Grid*

Xiangli Qu¹, Xue-Jun Yang¹, Guang Xiang², and Haifang Zhou¹

¹ School of Computer Science, National University of Defence Technology
Changsha, China, 410073
cathysmile@eyou.com

² Institute of Computer Software and Theory, Northeastern University
Shenyang, China, 110004

Abstract. Currently, credentials are widely used in attribute/property-based trust establishment. Yet, for the scale and openness of Grid, much uncertainty comes with credentials, such as no globally accepted authority, no guarantee of the consistency between behaviors and claims etc., which is less studied by most credential-based approaches for trust establishment. In this paper we introduce the notion of credential trustworthiness to bring credentials' uncertainty and differentiation to the surface. Using uncertainty-reasoning methods, we give the evaluation of an entity's trustworthiness with one single credential involved and multiple credentials involved respectively.

1 Introduction

Grid technologies enable cooperation among a large scale of dynamic, heterogeneous, varied and distributed resources. With such a tremendous pool of resources, it is common for entities to interact with unknown correspondents from time to time. As a result, trust establishment in such scenarios becomes an extremely important issue for reliable Grid computing. For the scale and openness of Grid circumstances, traditional identity-based approaches are not effective. Currently, one emerging trend is the adoption of attribute/property-based approaches such as ATN (Automated Trust Negotiation) [1,2,3,4], which resort to credential exchanges to establish trust relationships. A credential describes one or more attributes of the owner, using attribute name/value pairs to represent properties of the owner asserted by the issuer.

In such credential-based approaches for trust establishment, an entity with some credentials will be regarded as a trusted cooperator. Current studies in this field mostly focus on negotiation strategies and sensitive attribute protection [1,2,3,4]. Yet, an important fact must not be overlooked for Grid circumstances: much uncertainty comes with credentials. As there is no universally trusted authority in Grid and interactions frequently occur across multiple domains, credentials signed by issuers from

* This work is partially supported by the National 863 High Technology Plan of China under the grant No. 2002AA1Z201, China 863 OS New Technologies Research under the grant No. 2003AA1Z2060 and the Grid Project sponsored by China ministry of education under the grant No. CG2003-GA00103.

external domains cannot be fully trusted. Even if credentials can be fully trusted, there is no guarantee that an entity with some credentials will act as expected. Meanwhile, different credentials will have differentiated strength in demonstrating to what degree its owner should be trusted in different context. Though some studies have explored the verification and revocation issues related to credentials [5], the effect of credential's uncertainty on trust evaluation is rarely quantitatively analyzed. Therefore, we introduce the notion of credential trustworthiness and borrow the idea of uncertainty reasoning from expert system [6,7] for trust evaluation, seeking to show the uncertainty and differentiation of credentials, meet the requirement that different trust evaluation should be made for entities with different credentials, and as a result, achieve an efficient and reliable trust establishment.

The rest of this paper is structured as follows: In section 2 some definitions used in our evaluation are introduced, based on which we give some lemmas; In section 3 and section 4, trust evaluation involving a single credential and multiple credentials are presented respectively; In section 5, a case study is provided; And finally in section 6, we conclude the paper.

2 Definitions

Some basic definitions to be used in our evaluation are listed as follows:

Definition 1: Trustworthiness, denoted by T ($0 \leq T \leq 1$), describes to what degree some facts can be trusted to be true or some actions can be trusted to occur as expected. The trustworthiness of a credential c , denoted by $T(c)$, shows to what degree one can believe that the attributes/properties asserted in the credential c are true. The trustworthiness of an entity e , denoted by $T(e)$, shows to what degree one can believe that the entity will act as expected.

Definition 2: Credential Trustworthiness Factor (CTF), shows to what degree a credential c will change the trustworthiness of an entity e , denoted by $CTF(e, c)$. Since a credential possessed or unpossessed will have opposite effect on an entity's trustworthiness, we divide this factor into two kinds: Credential Possessed Trustworthiness Factor ($CPTF$) and Credential Unpossessed Trustworthiness Factor ($CUTF$), which are defined in Definition 3 and Definition 4 respectively.

Definition 3: Credential Possessed Trustworthiness Factor, denoted by $CPTF(e, c)$, shows to what degree the possession of a credential c will increase the trustworthiness of an entity e , which is expressed in formula (1), where $p(e)$ ($0 < p(e) < 1$) stands for the prior probability that entity e can be trusted when it shows no credentials and $p(e|c)$ stands for the posterior probability that entity e can be trusted after it is verified to possess credential c :

$$CPTF(e, c) = \frac{p(e|c) - p(e)}{1 - p(e)} \quad (1)$$

Since $p(e|c) \geq p(e)$, the value of $CPTF(e, c)$ is between 0 and 1. And from formula (1), we can easily get:

$$CPTF(e, c) = \frac{p(\neg e) - p(\neg e|c)}{p(\neg e)} \quad (2)$$

As $p(\neg e)$ can be deemed as entity e 's untrustworthiness with no credential showing and $p(\neg e|c)$ can be deemed as entity e 's untrustworthiness when it is verified that e possesses credential c , $CPTF(e, c)$ can be seen as the decreasing rate of entity e 's untrustworthiness after verifying that it possesses credential c . Therefore, the bigger the value of $CPTF(e, c)$, the greater decrease in entity e 's untrustworthiness, and the greater increase in entity e 's trustworthiness.

Lemma 1:

$$p(e|c) = CPTF(e, c) + (1 - CPTF(e, c)) \times p(e) \quad (3)$$

Proof. From Definition 3, it can be easily proved. Therefore, with $p(e)$ and $CPTF(e, c)$, we will get the value of $p(e|c)$.

Definition 4: Credential Unpossessed Trustworthiness Factor, denoted by $CUTF(e, \neg c)$, shows to what degree the unpossession of a credential c will decrease the trustworthiness of an entity e , which is expressed in formula (4):

$$CUTF(e, \neg c) = \frac{p(e|\neg c) - p(e)}{p(e)} \quad (4)$$

Since $p(e|\neg c) \leq p(e)$, the value of $CUTF(e, \neg c)$ is between 0 and -1. And from formula (3), we can easily get:

$$CUTF(e, \neg c) = -\frac{p(e) - p(e|\neg c)}{p(e)} \quad (5)$$

From formula (5), we can see $CUTF(e, \neg c)$ expresses the decreasing rate of entity e 's trustworthiness after verifying that e does not possess credential c . Obviously, the lesser the value of $CUTF(e, \neg c)$, the greater decrease in entity e 's trustworthiness.

Lemma 2:

$$p(e|\neg c) = (1 + CUTF(e, \neg c)) \times p(e) \quad (6)$$

Proof. From Definition 4, the proof can be easily got. Therefore, with $p(e)$ and $CUTF(e, \neg c)$, we will get the value of $p(e|\neg c)$.

Definition 5: Credential Trustworthiness Factor Eigenvector, denoted by $CTFE(e, c)$, includes three eigenvalues: $((T(c), CPTF(e, c), CUTF(e, \neg c)))$.

3 Single Credential-Based Trust Evaluation

With the above definitions, now we will focus on specific trust evaluation based on CTFE. First, we will begin with single credential-based evaluation.

From Bayes Theory, we can get the following equation:

$$p(e) = p(e|c) \times p(c) + p(e|\neg c) \times p(\neg c) \quad (7)$$

Using Lemma 1 and Lemma 2, $p(e|c)$ and $p(\neg e|c)$ can be replaced with $CPTF(e, c)$ and $CUTF(e, \neg c)$, therefore formula (7) will change to:

$$p(e) = (CPTF(e, c) + (1 - CPTF(e, c)) \times p(e)) \times p(c) + (1 + CUTF(e, \neg c)) \times p(e) \times p(\neg c) \quad (8)$$

From formula (8), we can get the following lemma:

Lemma 3:

$$p(c) = \frac{CUTF(e, \neg c) \times p(e)}{CUTF(e, \neg c) \times p(e) - CPTF(e, c) \times (1 - p(e))} \quad (9)$$

Formula (9) shows that: with $CTFE(e, c)$, we can calculate the prior possession probability $p(c)$ of credential c .

From probability formula, we get:

$$p(e|s) = p(e|c) \times p(c|s) + p(e|\neg c) \times p(\neg c|s) \quad (10)$$

Wherein s stands for all observations related to credential c .

Next, we will consider three special scenarios for formula (9):

(1) When $p(c|s) = 1$, i.e., credential c is absolutely trusted, we can get:

$$p(e|s) = p(e|c);$$

(2) When $p(c|s) = 0$, i.e., credential c is absolutely untrusted, we can get:

$$p(e|s) = p(e|\neg c);$$

(3) When $p(c|s) = p(c)$, from Bayes Theory we can get:

$$p(e|s) = p(e|c) \times p(c) + p(e|\neg c) \times p(\neg c) = p(e);$$

For the other scenarios, we can get the value of $p(e|s)$ according to Subsection Linear Interpolation Formula, to summarize:

$$p(e|s) = \begin{cases} p(e|\neg c) + \frac{p(e) - p(e|\neg c)}{p(c)} \times p(c|s), & 0 \leq p(c|s) < p(c) \\ p(e) + \frac{p(e|c) - p(e)}{1 - p(c)} \times [p(c|s) - p(c)] & p(c) \leq p(c|s) \leq 1 \end{cases} \quad (11)$$

Since $T(c)$ can be seen as the probability that we can trust credential c , therefore we can use $T(c)$ to estimate $p(c|s)$ and get the following lemma:

Lemma 4:

$$p(e|C_{T(c)}) = \begin{cases} p(e|\neg c) + \frac{p(e) - p(e|\neg c)}{p(c)} \times T(c), & 0 \leq T(c) < p(c) \\ p(e) + \frac{p(e|c) - p(e)}{1 - p(c)} \times [T(c) - p(c)] & p(c) \leq T(c) \leq 1 \end{cases} \quad (12)$$

From Lemma 1, Lemma 2 and Lemma 3, we know that $p(e|c)$, $p(e|\neg c)$ and $p(c)$ can all be calculated with $CTFE(e, c)$. Therefore, given $CTFE(e, c)$, with a single credential c involved, entity e 's trustworthiness $T(e) = p(e|C_{T(c)})$ can be evaluated according to formula (12).

4 Multi-credential-Based Trust Evaluation

For multi-credential-based trust evaluation, we will only consider n credentials $c_1, c_2 \dots c_n$ with mutually independent effect on an entity e 's trustworthiness and untrustworthiness. We have the following theorem:

Theorem 1:

$$p(e|c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}) = \frac{p(e|c_{1T(c_1)})p(e|c_{2T(c_2)}) \dots p(e|c_{nT(c_n)})p(\neg e)^{n-1}}{p(\neg e|c_{1T(c_1)})p(\neg e|c_{2T(c_2)}) \dots p(\neg e|c_{nT(c_n)})p(e)^{n-1} + p(e|c_{1T(c_1)})p(e|c_{2T(c_2)}) \dots p(e|c_{nT(c_n)})p(\neg e)^{n-1}} \quad (13)$$

Proof:

Since $c_1, c_2 \dots c_n$'s influence to e 's trustworthiness and untrustworthiness are mutually independent, therefore:

$$\frac{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}|e)p(e)}{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}|\neg e)p(\neg e)} = \frac{p(c_{1T(c_1)}|e)p(c_{2T(c_2)}|e) \dots p(c_{nT(c_n)}|e)p(e)}{p(c_{1T(c_1)}|\neg e)p(c_{2T(c_2)}|\neg e) \dots p(c_{nT(c_n)}|\neg e)p(\neg e)} \quad (14)$$

From Bayes Theory, we know $p(e|c) = \frac{p(c|e) \times p(e)}{p(c)}$, therefore:

$$p(e|c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}) = \frac{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}|e)p(e)}{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)})} \quad (15)$$

$$p(\neg e|c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}) = \frac{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)}|\neg e)p(\neg e)}{p(c_{1T(c_1)}c_{2T(c_2)} \dots c_{nT(c_n)})} \quad (16)$$

As $p(c|e) = \frac{p(e|c)p(c)}{p(e)}$ and $p(c|\neg e) = \frac{p(\neg e|c)p(c)}{p(\neg e)}$, from formula,

(14), (15), (16), we can easily get:

$$\begin{aligned} \frac{p(e|c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)})}{p(\neg e|c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)})} &= \frac{p(c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)}|e)p(e)}{p(c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)}|\neg e)p(\neg e)} \\ &= \frac{p(c_{1T(c_1)}|e)p(c_{2T(c_2)}|e)\dots p(c_{nT(c_n)}|e)p(e)}{p(c_{1T(c_1)}|\neg e)p(c_{2T(c_2)}|\neg e)\dots p(c_{nT(c_n)}|\neg e)p(\neg e)} \\ &= \frac{p(e|c_{1T(c_1)})p(e|c_{2T(c_2)})\dots p(e|c_{nT(c_n)})p(e)^{n-1}}{p(\neg e|c_{1T(c_1)})p(\neg e|c_{2T(c_2)})\dots p(\neg e|c_{nT(c_n)})p(\neg e)^{n-1}} \end{aligned} \quad (17)$$

Therefore,

$$\begin{aligned} p(e|c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)}) &= \\ &= \frac{p(e|c_{1T(c_1)})p(e|c_{2T(c_2)})\dots p(e|c_{nT(c_n)})p(e)^{n-1}}{p(\neg e|c_{1T(c_1)})p(\neg e|c_{2T(c_2)})\dots p(\neg e|c_{nT(c_n)})p(\neg e)^{n-1} + p(e|c_{1T(c_1)})p(e|c_{2T(c_2)})\dots p(e|c_{nT(c_n)})p(e)^{n-1}} \end{aligned} \quad (18)$$

And Theorem 1 is proved.

As presented in section 4, given $CTFE(e, c_i)$, $p(e|c_i)$ and $p(\neg e|c_i)$ can both be calculated ($i = 1, 2, \dots, n$). Consequently, $p(e|c_{1T(c_1)}c_{2T(c_2)}\dots c_{nT(c_n)})$ can also be calculated. In other words, credential e 's trustworthiness with n credentials involved can be evaluated.

5 A Case Study

Based on ratings from others, an entity e 's prior trustworthiness $p(e)$ is estimated as 0.6, and the $CTFE$ of 2 credentials c_1 and c_2 are:

$$CTFE(e, c_1) = (0.5, 0.6, -0.3), \quad CTFE(e, c_2) = (0.1, 0.8, -0.1)$$

From Lemma 1, we get:

$$p(e|c_1) = 0.84, \quad p(e|c_2) = 0.92$$

From Lemma 2, we get:

$$p(e|\neg c_1) = 0.42, \quad p(e|\neg c_2) = 0.54$$

From Lemma 3, we get:

$$p(c_1) = 0.429, \quad p(c_2) = 0.158$$

From Lemma 4, we get:

$$p(e|c_{1T(c_1)}) = 0.630, \quad p(e|c_{2T(c_2)}) = 0.578$$

That is to say: Only considering credential c_1 , entity e 's trustworthiness $T(e)$ is evaluated as 0.630; Only considering credential c_2 , entity e 's trustworthiness $T(e)$ is evaluated as 0.578.

From Theorem 1, we get:

$$p(e | C_{1T(c_1)} C_{2T(c_2)}) = 0.609$$

That is to say: considering credential c_1 and c_2 , entity e 's trustworthiness $T(e)$ is evaluated as 0.609.

6 Conclusions

In this paper, we focus on the uncertainty and differentiation of credentials in credential-based trust establishment. With the introduction of credential trustworthiness factor eigenvector, we give the evaluation of an entity's trustworthiness with one single credential involved and multiple credentials involved respectively. In the overall evaluation, uncertainty-reasoning methods are adopted. We believe, this kind of trustworthiness evaluation will benefit a more efficient and reliable trust establishment.

References

1. Ting Yu, Automated Trust Establishment in Open Systems, Ph.D thesis, University of Illinois, 2003.
2. W. Winsborough, K. Seamons, and V. Jones, Automated Trust Negotiation. In DARPA Information Survivability Conference and Exposition, Hilton Head Island, SC, Jan. 2000.
3. R. Jarvis, Selective Disclosure of Credential Content during Trust Negotiation. Master's thesis, Dept. of Computer Science, Brigham Young University, Apr. 2003.
4. W. Winsborough and N. Li. Protecting Sensitive Attributes in Automated Trust Negotiation. In ACM Workshop on Privacy in the Electronic Society, Washington, DC, Nov. 2002.
5. P. Kocher. On Certificate Revocation and Validation. In International Conference on Financial Cryptography, Anguilla, British West Indies, Feb. 1998.
6. Jun Ma, Jun Liu and Yang Xu, A Method of Uncertainty reasoning by using information. Proceedings of 31st IEEE International Symposium on Multiple-Valued Logic:373-377, 2001.
7. Jensen F. V., An Introduction to Bayesian Networks, Springer Verlag, New York, NY, 1996.

Towards the Merger of Grid and Economy

Thomas Weishäupl and Erich Schikuta

Department of Computer Science and Business Informatics
University of Vienna
Rathausstraße 19/9, A-1010 Vienna, Austria
{thomas.weishaeupl,erich.schikuta}@univie.ac.at

Abstract. This paper presents a discussion on Grids merging with the economy. Until now no clear notion and categorization exist in this area. Problem formulations of “economy grid” projects have difficulties in lingual descriptions, and are going with a hype based on vague assumptions and therefore weak problem and requirement specifications. In this paper we present the economy-grid (EG) layer model to clarify many interacting aspects. The model results from historical observations and is proven by applying it to the recent Grid projects and initiatives.

1 Introduction

The economy is an old, historical, sociological, and mature system influencing many areas of human life. Science describes and analyzes this system by the research areas of political economics [1] and business economics. Economy is a driving factor which motivates technological developments out of others as human needs [2], both based on humans yearning [3].

Conclusively also Grid computing has an interdependency with economy. The term “Grid” describes a distributed computing infrastructure (see Sec. 2), which establishes *sharing* of resources and not only information (as the Internet does).

The Grid is based on Internet technology and was initiated by the research community. However like the Internet today, applications and new business models are imaginable. One the one hand various different ideas and concepts from the economy are used in Grid technology. On the other hand the economy has also requirements for applications in his area. Figure 1 shows the relationship and mutual inputs providing to each other.

The structure of the paper is as follows. Section 2 defines the scope of our work and presents some definitions for clarification. In Section 3 the motivation for our approach is given. We explain in Section 4 which leads to the Economy Grid - Layer Model of Section 5. Section 6 justifies the layer model and presents projects, software, concepts and other initiatives containing economy and Grid aspects. The paper is finished by a summary and some conclusions are drawn.

2 Scope and Definitions

To avoid confusions we define our scope and underline our point of view on economy, E-Commerce, Business Model, and Grid, respectively.

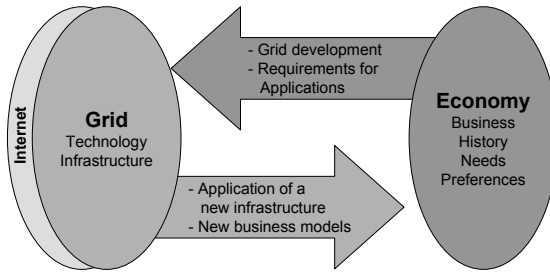


Fig. 1. Relation between Economy and the Grid

Economy The term *economy* is quite common and evident. In our work we use the term to express activities related to the production and distribution of goods and services in a particular environment. With the adjective “economic” we understand the correct and cost effective use of available resources.

E-Commerce We understand and use the term *Electronic Commerce* (“E-Commerce”) in our work as conducting business communication and transactions over networks and through computers. As most restrictively defined, electronic commerce is the buying and selling of goods and services, and the transfer of funds, through digital communications. E-Commerce also includes doing business online, typically via the Web. Although in most cases E-Commerce and E-Business are synonymous, E-Commerce implies that goods and services can be purchased online, whereas E-Business might be used as more of an umbrella term for a comprehensive commercial presence on the Web.

Business Model A *Business Model* describes the operations of a business including the components of the business, the functions of the business, and the revenues and expenses that the business generates.

Grid A *Grid* is a type of parallel and distributed system that enables the *sharing* of geographically distributed “autonomous” resources dynamically at runtime depending on their availability, capability, performance, cost, and users’ quality-of-service requirements. By this, virtual organizations can be established [4].

3 Motivation

This section focus and points out the requirements and needs from different points of view, as economy, E-Commerce, and Grid developments, respectively.

Economy Varian [1, chap. 34] describes the behavior of information technology and its market, where a new form of good can be traded over the Internet. These “new” goods show different properties to common goods (e.g. cars, food, etc.), as no efforts for transport and reproduction are necessary.

As the Internet gives new possibilities for E-Commerce, and supports individuals (companies, single person) to do their business, we believe that the Grid can extend these possibilities by far (see [4, Sec. 2]).

Firstly, the information technology resources in one homogenous organization are used more efficiently by a network. The Grid is developed to enhance resource sharing over networks even more. A maximized utilization of resources brings economical advantages.

Secondly, there are business opportunities for dynamic collaborations, new project workflows, software on demand, dynamic resource-management, resource on demand, application service providers, and other Grid information society components. The trading and accounting concepts for Grid resources will become crucial in this area.

Individual needs deliver new technological applications with specific requirements. Also, the economic growth is stimulated by the Grid. Therefore, today's research activities initiate an economy-grid interaction.

E-Commerce E-Commerce is a new type of doing commerce and business by the infrastructure of the Internet. Goods and services are promoted, marketed and distributed by new business models. Digital money is developed to allow this (see [5]).

But still the dissemination is limited by technical and sociological problems. Ketterer and Stroborn [6] describe them respectively, Security (Protection of personal data and confidentiality, Authentication, No disclaiming of an order or delivery, Minimising loss of payments), Usability, Legal environment and circumstances, Business models (hen-egg problem, payment method, etc.), and Customer loyalty.

Out of these reasons typical problems of E-Commerce show up, as the so called "Internet bubble" [7], and "new economy".

The Grid is based on Internet technologies. These are TCP/IP, HTTP, XML, and others, which are the bases of Grid protocols and services. E-Commerce based on Grid infrastructure can have less weak characteristics as the state of the art E-Commerce' once. E-Commerce infrastructure uses WWW clients combined with proprietary, complicated, or insecure payment methods.

Grid The Grid community is highly motivated to include economic aspects in its efforts.

Firstly, Grid technology can provide a new infrastructure for applications in institutions and can be sold in the future as an "out-of-the-box" product. There already exist some spin-off companies, as Avaki¹, Entropia² or others. Secondly, some Grid infrastructure components can use economic principles to accomplish their requirements. Optimization problems have to be solved. Many optimization concepts exit in the economy, as auctions and open markets to negotiate an optimal prise for goods. Also the optimal distribution of data or the optimal alignment of computing work have to be determined. Algorithms implementing economic concepts find quasi-optimal solutions for Grid problems (see [8, 9]).

¹ <http://www.avaki.com>

² <http://www.entropia.com>

Thirdly, in a Grid the resources can not be free, but accessible under certain user constraints. A market can regulate resource sharing in a satisfying way for a resource provider and customer. This can establish an open Grid, with similar properties as the Internet now for information.

Finally, an economic focused investigation of Grid technology can motivate the development resulting in financial investment pushing the Grid research activities.

4 Methodology

In this section we introduce levels or layers, which we use in the following to categorize recent Grid projects and other initiatives focusing on economy in general. The model derives from our observation of the technologic-economic developments of the Internet and the Grid. We show that all projects listed in Section 6 are classifiable in the layer model. However we do not pretend that the list is exhaustive.

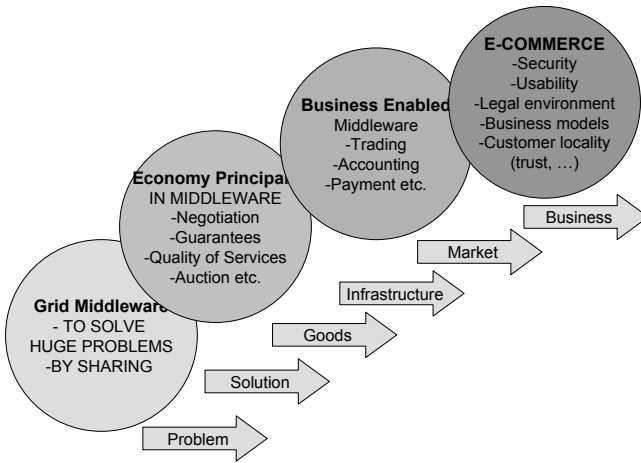


Fig. 2. Grid development to business

From our point of view Figure 2 visualizes the procedure of the economic development of technology. As a concrete example the evolution of the Internet and telephony can be named. Considering the Internet, which was invented to solve communication challenges in the area of military defence, it is now a communication infrastructure for everybody, used also for doing commerce between persons (e.g. Amazon, Ebay). Obviously out of a problem and its solution arises a new good. This good can be traded in an adapted infrastructure, which creates a market and so a new commercial business. An other example for this process is the evolution and usage of the telephone infrastructure.

The cycles of Figure 2 visualize a possible evolution of Grid technology. Grid middleware was developed to solve huge computational problems by sharing of resources inside of a virtual organization, e.g. the DataGrid project enables access to geographically distributed computing power and storage facilities belonging to different institutions. This will provide the necessary resources to process huge amounts of data coming from scientific experiments.

The Grid middleware uses not only principles and concepts of computer science, but also principles of the economy to provide the capabilities mentioned above, e.g. [9]. This middleware is a new good or product. Until now, the developments are in progress and no finalized “business enabled” middleware exists (e.g. Globus) , which can establish a new market or field of commerce.

5 Economy Grid – Layer Model

We propose an economy-grid layer (EG-layer) model, to better understand the problems in context of using Grid infrastructure as a new good or product. The model results from the observations and conclusions mentioned above.

The EG-layer model consists of four layers with the following characteristics:

EG1 Integration Layer: Grid Using Economic Principles Economic principles, concepts, and experience are integrated into the Grid and influence *developments* of Grid infrastructure, e.g. resource usage can be optimized by adaption of auction principles.

EG2 Commercialization Layer: Selling Grid Software Companies creating a product for “homogenous” organizations using Grid software or some components. They *sell* the recent open source software combined with self-developed software modules and services.

EG3 Enabling Layer: Business Enabled Grid The business enabled Grid establishes an open Grid, with similar properties as the Internet for information today. An infrastructure for a market has to be provided. In a Grid the resources can not be free, but accessible under user constraints. A market can regulate resource sharing in a satisfying way for a resource provider and customer. A *single business* needs trading, accounting and payment mechanisms.

EG4 Modelling Layer: Business Models on Grid The market enabled Grid infrastructure gives possibilities for new business models and *E-Commerce*.

Figure 3 shows the context of the EG-layer model with grid development progress, time and economic usage. The recent research work is done on the first three layers only. Higher layer depends on lower layer, whereas the light-gray vertical bars represent the quantity of dependency. Layer 1 interacts with Layer 3, because e.g. about a price of a resource of Layer 3 has to be agreed by a resource allocation mechanism (broker, scheduler) of Layer 1. All layers are necessary to establish the economic usage of a Grid in the future. The difference between Layer 3 and Layer 4 is, that Layer 3 infrastructure provides business between known partners. The Layer 4 infrastructure establishes sophisticated business models and highly dynamic virtual organizations.

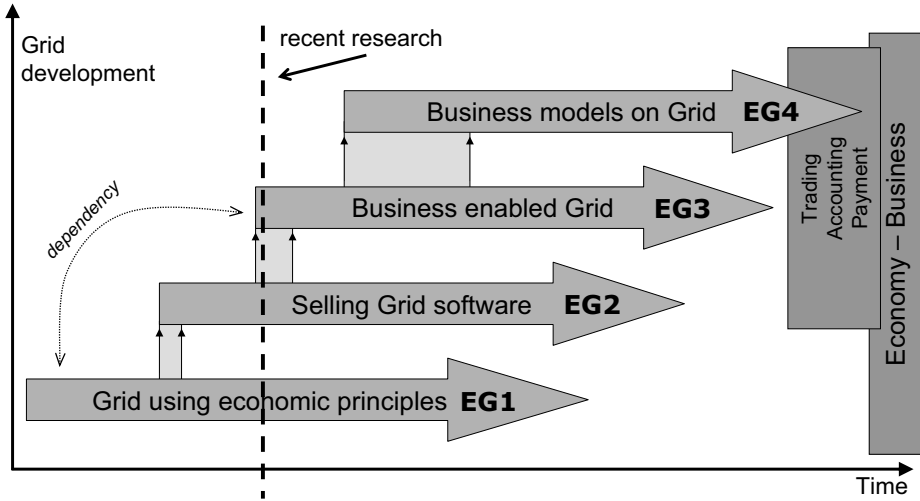


Fig. 3. Economy-grid layer (EG-layer) model

6 Categorization of Grid Research Based on the EG Layer Model

This section applies the EG-layer model to categorize a few approaches in the Grid community, which focus on economic aspect on different layer of abstraction. The project are only a few examples to confirm the EG model.

6.1 EG1 – Integration Layer: Grid Using Economic Principles

The first layer comprise Grid developments using economic principles. These optimization principles are used especially in the field of data management and work load management, respectively:

DataGrid WP2 Data Management Data has to be distributed and replicated. This is provided by a replica management system containing a replica optimization service, called *Optor*. The aim of *Optor* is to optimize the creation and deletion of the replicas. Until now, the long term optimization is implemented by *OptorSim* [9]. It evaluates different data distribution strategies and algorithms.

Gridbus The project uses economic-based distributed resource management and scheduling (see [8]). It proposes a Grid Architecture for Computational Economy (GRACE), which is realized by the resource broker *Nimrod-G*. The market architecture and scheduler has budget and cost notions using a simple heuristic minimization of costs and time under soft constraints represented by deadline or budget.

6.2 EG2 – Commercialization Layer: Selling Grid Software

The second layer of the EG model comprises all software projects or products, which can be used as Grid middleware software in a commercial environment, and are provided by a company.

Avaki The company Avakibrings to market the Grid middleware software Legion. The Legion software was designed and developed by a research project at the University of Virginia.

Entropia Entropia provides software to build desktop Grids, to establish, so called, PC Grid computing. The Entropia software enables the dynamic usage of idle processor cycles on desktop computers.

6.3 EG3 – Enabling Layer: Business Enabled Grid

EG3 describes different research projects with the goal to provide knowledge and software for business needs. Resources are not free in a Grid and only accessible under certain user constraints. By an economy Grid infrastructure it should be possible to manage these constraints effectively.

GRIA The GRIA project will develop, apply and evaluate a Grid testbed, based on an existing open-source infrastructure but incorporating services for end-to-end quality of service to provide reliable and manageable performance and support secure, end-to-end business models and processes, enabling the Grid to be used for outsourcing computational services.

GGF Grid Economic Services Architecture Working Group The goal of GESA-WG is to provide the supporting infrastructure to enable Computational and Data Grids operated by different organizations to “trade” services between each other.

6.4 EG4 – Modelling Layer: Business Models on Grid

The EG4 layer categorizes initiatives and projects which deal with E-Commerce, business models and market enabled Grid infrastructure. As described in Section 4, no implementation of corresponding models exist until now. An initiative of the EG4 layer is the *Business In the Grid Infrastructure (BIG)* [10] project. It has the goal to understand the possibilities and needs of doing business in the Grid infrastructure. To make the economy prepared for the upcoming Grid infrastructure, information work and demand analysis have to be done, allowing the development of novel business models stimulating the IT economy.

7 Conclusion

This paper gives an introduction to the recent research activities concerning Grid computing and the economy. The infrastructure development learns from

the economy to build the Grid and also the economy applies findings of the upcoming technology. Many aspects interact on different abstraction levels.

Grid technology passes the inception phase and no assured predictions of further development can be done. Nevertheless, it is possible that it will have similar effects on economy and the human life as the Internet, but until now the technology is not matured enough to provide this. On the other hand Grid technology can learn from the economy to develop better problem solution approaches.

The Grid is not the only “player in the field” tackling economic aspects. Web services are common place even stronger in the economic field. The development of Grid technology and Web services can benefit by the recent envisioned combination of both (as the WS-Resource Framework) and can provide even better services to commercial applications.

The most important aspect for a business model an the economy is the kind of good, which can be traded. The Quality of Service of a Grid has to be stabilized to get a real tradeable good. We see in this issue a strong impetus for further research.

Acknowledgements

The work described in this paper was supported by project number 10547 of the OeNB Anniversary Fund. We thank Heinz Stockinger for the fruitful discussions which improved the quality of the paper.

References

1. Varian, H.R.: *Intermediate Microeconomics*. 5th edn. W. W. Norton and Company, Inc., New York, USA (2000)
2. Maslow, A.: *Motivation and Personality*. Harper and Row, New York (1954)
3. Giussani, L.: *The Religious Sense*. McGill-Queens (1997)
4. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum (2002)
5. Lynch, D.C., Lundquist, L.: *Digital Money - The New Era of Internet Commerce*. John Wiley and Sons, USA (1996)
6. Ketterer, K.H., Stroborn, K.: *Handbuch ePayment: Zahlungsverkehr im Internet ; Systeme, Trends und Perspektiven*. Fachverlag Deutscher Wirtschaftsdienst, Koeln (2002)
7. Perkins, A.B., Perkins, M.C.: *The Internet Bubble*. HarperBusiness (1999)
8. Buyya, R.: *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. PhD thesis, School of Computer Science and Software Engineering, Manoash University, Melbourne, Australia (2002)
9. Bell, W.H., Cameron, D.G., Capozza, L., Millar, A.P., Stockinger, K., Zini, F.: *OptorSim - A Grid Simulator for Studying Dynamic Data Replication Strategies*. *International Journal of High Performance Computing Applications* **17** (2003)
10. Schikuta, E., Weishäupl, T.: *Business in the Grid infrastructure*. University of Vienna. (2004) <http://www.informatik.univie.ac.at/big>.

An Approach to Evaluate Communication Cost in Grid System

DaDong Wang, HongJun Wang, RuiJun Wang, HaiYing Liang, and Yuan Gao

School of Information Science & Engineering Northeastern University
110004 Shenyang, China
wdd@mail.neuq.edu.cn

Abstract. Resource usage cost is an important aspect of resource management in a grid system. This paper describes an approach to evaluate communication cost in terms of the IP address of participants. According to the way of the settlement in the Internet, we use autonomous system (AS) relationships and AS path to evaluate. We also present an approach to infer AS relationships from BGP routing tables.

Keywords: Communication cost, AS relationship, IP address, BGP, AS graph

1 Introduction

A grid is a very large-scale network computing system that scales to Internet size environments with machines distributed across multiple organizations and administrative domains[1]. A grid collects geographically distributed resource harnessed together to satisfy various needs of the users. Resources owned by various administrative organizations can be computers, storage space, software applications, and data. In a grid system resources are added and removed dynamically. Grid systems often deal with intermittent participation and highly variable behavior. In the case of Mojo Nation, it is reported that average connection time was only 28% and highly skewed(one sixth of the nodes always connected). Different types of applications are executed with different resources requirements. While some type of applications need high performance, some type of applications are required to constrain cost even if it means reduced performance. Therefore, the resource management framework should evaluate the resource usage costs and select parts of available resources to match user request. Communication cost is one part of the resource usage costs. Indeed, a number of systems have been built using a market mechanism to allocation the resources[7]. However, few of them paid attention to the communication cost. This paper describes an approach to evaluate communication cost in terms of the IP address of participants.

2 Communication Cost in the Internet

The Internet connects thousands of Autonomous Systems (ASes) operated by many different administrative domains. Routing between ASes is determined by

the interdomain routing protocol such as Border Gateway Protocol (BGP). An AS applies local policies to select the best route for each prefix and to decide whether to propagate this route to neighboring ASes, without divulging these policies or the AS's internal topology to others[2]. In practice, BGP policies reflect the commercial relationships between neighboring ASes. AS pairs typically have a provider-customer, peer-peer, sibling-sibling relationship[3]. A customer pays its provider for connectivity to the rest of the Internet. Therefore, a provider does transit traffic for its customers. A pair of peers agree to exchange traffic between their respective customers free of charge. A pair of siblings allows a pair of ASes to provide connectivity to the rest of the Internet for each other. We represent AS relationships by a graph G whose edges are either directed or undirected. Each vertex is an AS, a directed edge from vertex u to vertex v indicates that v is a customer of u , and an undirected edge indicates that u and v are peers or siblings. Figure 1 shows an example.

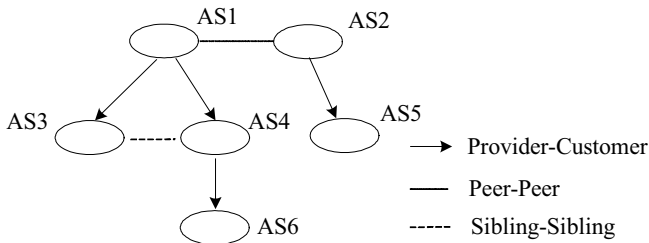


Fig. 1. As relationship Graph

When the traffic transits a boundary between a client and the provider, or between two providers, the settlement is performed. There are three kinds of financial settlements in the Internet: Sender Keep All (SKA), provider/customer role selection and negotiated financial settlement[4]. SKA peering arrangements are those in which traffic is exchanged between two or more ISPs without mutual charge. A customer funds its provider to complete the delivery through an interconnection mechanism. The simplest form of the third settlement is to measure the volume of traffic being passed in each direction across the interconnection and to use a single accounting rate for all traffic. The accounting rate can be negotiated to be any amount. The first and second settlement respectively correspond to the peer-peer and customer-provider relationship. As the accounting rate will have to match the traffic flow which is relative to the AS relationships, the third settlement is also relative to AS relationships.

Consider the settlement in figure 1. According to ASes relationships, when AS4 transits traffic for AS6 to other AS, AS6 pays AS4, but the traffic from AS1 to AS2 or from AS3 to AS4 does not need payment.

The underlying resource pool of a grid system is represented by a graph G , with n nodes and m edges, where each node u_i is an AS and contains $N(u_i)$ hosts, each edge (u_i, u_j) indicates a cost $C(u_i, u_j)$. We denote an edge from a customer to

a provider with $C(u_i, u_j)=1$, an edge between two peers/siblings with $C(u_i, u_j)=0$ and an edge from a provider to a customer with $C(u_i, u_j)=0$. Let p_{kl} denote the average unit communication cost between host H_k and host H_l . If $H_k \in u_i$, $H_l \in u_j$, u_i 's BGP routing table contains an entry with AS path($u_i \dots u_j$), we get

$$p_{kl} = \begin{cases} 0 & (u_i = u_j) \\ \sum_{x=i}^{j-1} C(u_x, u_{x+1}) & (u_i \neq u_j) \end{cases} \quad (1)$$

When users submit the application and input data along with some requirements to the resource management system, the resource management system should estimate the specific resource requirements for running the application. If the required resource does not exceed the available resource, the management system should select parts of resource to run the application. The resource management system should know which parts of resource have lower resource costs including the communication cost.

Suppose the resource management system has a IP address list of the available hosts. We can find the AS including certain host by searching a BGP routing table. We show several AS7018's BGP routing table entries below. We can see a AS maybe possess several prefixes which must not be sequential. For example, prefix 4.0.0.0/8 and prefix 8.0.0.0/8 belong to AS3356. Given a host 3.1.1.1, we can find it belongs to AS80.

Network	path
3.0.0.0/8	7018 80 i
4.0.0.0/8	7018 3356 i
4.17.225.0/24	7018 701 11853 6496 i
8.0.0.0/8	7018 3356 i

Suppose we know AS relationship on each edge. Using the formula (1), we can evaluate the average unit communication cost between two hosts.

3 Infer AS Relationship from BGP Routing Table

3.1 Basic Knowledge

BGP allows each AS to choose its own administrative policy in selecting routes and propagating reachability information to others. Each AS sets up its export policies according to its relationships with neighboring ASes. The AS relationships translate into the following rules that govern BGP export policies[4]:

In exchanging routing information with a provider, an AS can export its routes and its customer routes, but usually does not export its provider or peer routes. In exchanging routing information with a customer, an AS can export its routes and its customer routes, and as well as its provider or peer routes. In exchanging routing information with a peer, an AS can export its routes and its customer routes, but usually does not export its provider or peer routes. In exchanging routing information with a sibling, an AS can export its routes and routes of its customers, and as well as its provider or peer routes.

Export policies have a direct influence on the AS paths seen from a particular point in the Internet. If every AS sets its export policies according to

the above export rules, then no path should traverse a customer-provider edge after traversing a provider-customer or peer-peer edge, and no path would ever traverse more than one peer-peer edge. A valid AS path is composed of:

- (1) m customer-provider or sibling-sibling edges ($m \geq 0$)
- (2) k peer-peer edges ($k \leq 1$)
- (3) n provider-customer or sibling-sibling edges ($n \geq 0$)

A strictly hierarchical model of Internet structure is one in which a small number of global ISP transit operators is at the top, a second tier is of national ISP operators, and a third tier consists of local ISPs. At each tier the ISPs are clients of the tier above. In practice, although the boundary between second tier and third tier is ambiguous, the top tier is explicit.

Lemma 3.1 The AS relationships inferred from one AS's BGP routing table are not exclusive.

Proof: Suppose there is a AS path $(u_1, u_2, \dots, u_n)$ in u_1 's BGP routing table and u_2 is a customer of u_1 . No matter the inference shows that the relationship between u_1 and u_2 is peer-peer or provider-customer, the AS path is valid.

Lemma 3.2 A valid path replaced sibling-sibling edges with customer-provider edges (in the part one) or provider-customer edges (in the part three) is still valid. Theorem 3.1 Using a tier-one AS's BGP routing table to infer AS relationships, A inference error does not impact other AS relationships.

Proof: Suppose there is a AS path $(u_1, u_2, \dots, u_n)$ in a BGP routing table of certain tier-one AS. Consider the relationship between u_1 and u_2 . As u_1 is a tier-one AS, the relationship between u_1 and u_2 should be peer-peer or provider-customer. In a valid AS path, a provider-customer or peer-peer edge can be followed by only provider-customer or sibling-sibling edge. No matter the relationship between u_1 and u_2 is peer-peer or provider-customer, the relationship of u_i and u_{i+1} ($2 \leq i \leq n-1$) could only be provider-customer or sibling-sibling. Therefore, the erroneous inference of the AS relationship between u_1 and u_2 should not impact the relationship between u_i and u_{i+1} ($2 \leq i \leq n-1$). In the case of considering the relationship between u_i and u_{i+1} ($2 \leq i \leq n-1$), a similar argument applies.

Theorem 3.1 If there is a AS path $(u_1, u_2, \dots, u_n)$ and no AS path $(\dots u_{n-1}, u_n, u_{n+1} \dots)$ in a BGP routing table, the relationship between u_{n-1} and u_n can not be peer-peer or customer-provider.

Proof: We proof by contradiction. Suppose u_{n-1} is a peer of u_n . As u_n export its routes and its customer routes to u_{n-1} , there is at least one route which can reach a customer of u_n . This means that there is a AS path $(\dots u_{n-1}, u_n, u_{n+1} \dots)$. However, this contradicts the assumption. Therefore, u_{n-1} can not be a peer of u_n . Suppose u_{n-1} is a customer of u_n . As u_n export its routes, its customer's routes, its provider's routes and its peer's routes to u_{n-1} , there is at least one route which can reach a customer or a provider of u_n . This means that there is a AS path $(\dots u_{n-1}, u_n, u_{n+1} \dots)$. However, this contradicts the assumption. Therefore, u_{n-1} can not be a customer of u_n .

Let $Rasn_{uv}$ denote the number of the reachable ASes which pass edge uv for all BGP route.

Corollary 3.1 If $Rasn_{uv}=1$, u is a provider or a sibling of v .

3.2 Inference Rule

As the structure of Internet topology is complex and redundant, several tier-one ASes's BGP routing tables can not cover all the edge. We use BGP routing tables which belong to different tier to infer AS relationships.

```

For each AS path( $u_1, u_2, \dots, u_n$ ) in routing tables
  For each  $i=1, \dots, n-1$ 
    Neighbor[ $u_i$ ]=neighbor[ $u_i$ ]  $\cup$  { $u_{i+1}$ }
    Neighbor[ $u_{i+1}$ ]=neighbor[ $u_{i+1}$ ]  $\cup$  { $u_i$ }
    Ras[ $u_i u_{i+1}$ ]=Ras[ $u_i u_{i+1}$ ]  $\cup$  { $u_n$ }
  For each as  $u$ 
    Outdegree[ $u$ ]=|Neighbor[ $u$ ] |
  For each edge  $uv$ 
    Rasn[ $uv$ ]=|Ras[ $uv$ ] |
  To AS list sorted on outdegree:
  For each as  $i=1, 2, \dots, n$ 
    For each as  $j=i+1, \dots, n$ 
      If  $j$  not in {Neighbor[ $i$ ]}
        tierone[ $j$ ]=false
  For each AS path( $u_1, u_2, \dots, u_n$ )
    Find  $u_i$  whose outdegree is bigger than others
    For each as  $j=2, \dots, i$ 
      rank[ $u_j$ ]=max{rank[ $u_j$ ], rank[ $u_i$ ]+ $j$ }
    For each as  $j=i+1, \dots, n-1$ 
      rank[ $u_j$ ]=max{rank[ $u_j$ ], rank[ $u_n$ ]+ $n-j$ }

```

Fig. 2. Compute outdegree, $Rasn_{uv}$, tier-one AS, AS rank

The outdegree of a AS is an approximate measure of its size[8]. A small number of ASes form the transit core of Internet. They are so-called tier-one AS. In practice, the term "tier-one AS" is defined as an AS that does not have any upstream provider. The outdegrees of these ASes are larger than the other ASes's. There are dozens of tier-one ASes and some of them form a clique. We select these ASes belonging to the clique to infer AS relationships. Figure 2 shows a algorithm computing outdegree, the number of edge reachable ASes and tier-one AS.

(1) AS relationship priority

Consider a valid AS path ($u_1, u_2, \dots, u_n$). If the relationship between u_1 and u_2 could be peer-peer, provider-customer, or sibling-sibling, we select the peer-peer relationship. If the relationship between u_1 and u_2 could be provider-customer or sibling-sibling, we select the provider-customer relationship.

Using several BGP routing tables, we can verify a peer-peer relationship. Figure 3 shows a example. AS7018 and AS2914 are tier-one ASes. According to the valid AS path (1), we know that AS9318 is a peer or a customer of AS 7018, AS9457 is a customer or a sibling of AS9318, AS17589 is a customer or a sibling of AS9457. We select the relationship sequence (1). According to the valid AS path (2), we get the relationship sequence (2). Compared the sequence (1) with (2), we can infer that AS9318 is a customer of AS7018.

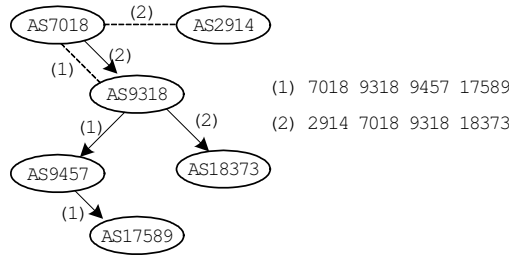


Fig. 3. Infer AS relationships from two BGP routing tables

Using several AS paths, we can verify a sibling-sibling relationship. Consider the following two AS paths: (1) 7018 3561 7474 7570 (2) 7018 101 7570 7474. According to the AS path (1), we get AS7474 is a provider of AS7570. According to the AS path (2), we get AS7570 is a provider of AS7474. Therefore, we infer that AS7474 is a sibling of AS7570.

(2) Treat valid AS paths

Consider a valid AS path $(u_1, u_2, \dots, u_n)$. If $u_1 \in \{\text{tier-one AS}\}$, we set u_2 is a peer of u_1 . If $u_1 \ni \{\text{tier-one AS}\}$ and $u_i \in \{\text{tier-one AS}\}$ ($1 < i < n$), we set the u_i 's neighbor whose outdegree is larger than another neighbor's is a peer of u_i . If $u_n \in \{\text{tier-one AS}\}$, we set u_{n-1} is a peer of u_n . If $u_i \ni \{\text{tier-one AS}\}$ ($1 \leq i \leq n$), we find the highest outdegree AS u_j , let u_j be the top provider of the AS path and set the u_j 's neighbor whose outdegree is larger than another neighbor's is a peer of u_j . Obviously, some inferences probably are not correct. Compared several results inferred from different BGP routing tables, we can verify these errors.

(3) Treat unusual AS paths

BGP export policies misconfiguration will result in unusual routes. For example, if a customer export its another provider route to its provider, it would result in a AS path which has a provider-customer edge followed by a customer-provider edge. As the outdegree of a AS is an approximate measure of its size, usually a provider's outdegree is large than its customer's outdegree. Consider a AS path $(u_1, u_2, \dots, u_n)$. Let Ou_{u_i} denote the outdegree of AS u_i , $\alpha = \min\{Ou_{u_{n-1}}/Ou_{u_n}, Ou_{u_{n+1}}/Ou_{u_n}\}$. If $\alpha \gg 1$, the AS path is a unusual path. Unusual paths will generate inference errors. We should delete them from BGP routing tables.

(4)AS rank

Figure 3 shows a algorithm to assign a rank to each AS. If the rank of a AS does not equal the rank of its peer, the relationship between this pair of ASes should be reassigned as provider-customer relationship.

3.3 Experimental Results

Use public available BGP routing table to infer AS relationships. We use ten BGP routing tables collected by RIPE NCC on Jan. 31,2004(<http://data.ripe.net>). Table 1 shows these ASes. There are 1284279 routing entries in these tables. These routing entries cover 16702 ASes and 31023 edges. Using the algorithm in Figure 3, we compute that AS 701, 7018, 1239, 209, 3356, 3549 and 2914 form a clique.

Table 1. RIPE NCC's peering ASes

AS	Name	Tier-one	Outdgreee	edges
513	CERN		61	21169
1103	SURFnet		156	21223
2914	Verio	Y	627	18709
3333	RIPE NCC		138	21519
3549	Globalcrossing	Y	688	20837
4608	Telstra		25	20811
4777	APNIC Pty Ltd		45	21384
7018	AT&T	Y	1719	20707
9177	Nextra		54	21086
13129	GAT		269	20802

Usually, there are hundreds of prefixes impacted by export policy misconfiguration[6]. Let $\alpha=16$, there are 452 BGP routing entries having unusual AS path. We delete them from routing table. We infer that there are 316 peer-peer edges, 30457 provider-customer edges and 228 sibling-sibling edges.

Although there is no publicly available information about AS relationships, we verify our inferred relationships by comparing with the results of other similar algorithms. [3] shows as much as 99.1% of the inference results are confirmed by the AT&T internal information (2000/3/9). Using the same BGP routing tables, the inference results of [3] show that AT&T have 15 peers and 1704 customers. Our inference results show that AT&T have 26 peers and 1693 customers. Among these peers, 13 peers are consistent with the results of [3], 7 peers are confirmed by the results of [5] and each remainder has a particular larger outdegree. This represents that our inference results are reliable.

As each BGP routing table has so many routing entries, it is not practical that the resource management system store all BGP routing entries for each AS. In fact, we can construct a BGP routing table for each AS from the AS relationship graph which annotated with the reachable customer prefixes on each provider-customer edge. In the same time, the hosts used by a grid system usually only

cover a small number of ASes. If we reserve the AS relationship graph and prefix lists of each AS, it is easy to generate the needed routing entries for each covered AS from the graph. Therefore, the resource management system should only reserve a AS relationship graph and prefix lists of each AS rather than all the BGP routing entries of each AS.

4 Conclusion

The relationships between ASes has a significant impact on evaluating the communication cost in a grid system. Our work makes two contributions toward evaluating the communication cost in terms of these relationships:

An approach for evaluating the communication cost based on IP address and AS relationships. An algorithm for inferring AS relationships from BGP routing tables. As the structure of the Internet is complex and the settlement model of the Internet is diverse, our approach has certain limitations: The evaluation is valid only when the cost that a packet paid to different transit AS is comparable. We infer AS relationships from a small number of BGP routing tables which can not cover all edges.

Despite these limitations, we have shown that our approach provides a view of the communication cost in terms of IP addresses, routes and AS relationships.

References

1. Krauter, K., Buyya, R., and Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. *Softw., Pract. Exper.* 32(2): 135-164 (2002)
2. Rekhter, Y., Li, T.: RFC 1771: A Border Gateway Protocol [S]. March 1995
3. Gao, L.: On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 2001, Dec: 733-745
4. Huston, G.: Interconnection, peering and settlements [J]. *Internet Protocol Journal*, 1999, 23(3): 45-51
5. Subramanian, L., Agarwal, S. and Rexford, J.: Characterizing the Internet hierarchy from multiple vantage points. *IEEE INFOCOM 2002*, 2002
6. Mahajan, R., Wetherall, D. and Anderson, T.: Understanding BGP Misconfiguration. *ACM SIGCOMM 2002*
7. Abramson, D., Buyya, R., and Giddy, J.: A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. *FGCS Journal*, 2002, 18(8): 1061-1074
8. Govindan, R. and Reddy, A.: An Analysis of Internet Inter-Domain Topology and Route Stability. In *Proc. IEEE INFOCOM'97*, 1997, 4

Towards the Automation of Autonomic Systems

Walid Chainbi

ENIS, Département d'Informatique, B.P. W
3038 Sfax, Tunisia
Walid.Chainbi@lycos.com

Abstract. The managing of today's computing systems goes beyond the administration of individual software environments. The need to integrate several heterogeneous environments into corporate-wide computing systems, and to extend that beyond company boundaries into the Internet, introduces new levels of complexity. Autonomic computing is considered as a promising solution to such problem. In this paper, we support the idea that agent architectural concepts are important to build such systems. We view autonomic elements as agents and autonomic systems as multi-agent systems. Although, there seems to be no commonly agreed properties that characterize an agent, there is a general consensus that autonomy is central to the notion of agency. In this paper, we propose a Petri nets with objects based formalism called cooperative objects for modeling autonomous agent systems. We mainly show how the proposed formalism favor autonomy property to a great extent.

1 Introduction

Recently, an interest has been witnessed in computing community to autonomic computing. Inspired by the functioning of the human nervous systems, autonomic computing is to design and build computing systems that possess inherent self-managing capabilities. Each autonomic system is a collection of autonomic elements - individual systems constituents that contain resources and deliver services to humans and other autonomic elements [1].

Many architectures are relevant to autonomic computing among which are intelligent agents, and multi-agent systems. In this paper, we support the idea that autonomic elements can be seen as agents and autonomic systems as multi-agent systems. An agent is a computational entity that can be viewed as perceiving and acting upon its environment and that it is autonomous in that its behavior at least partially depends on its own experience. A multi-agent system is a system designed and implemented as several interacting agents [2].

The agent technology community has made substantial progress in recent years in providing a theoretical and practical understanding of many aspects of agents and multi-agent systems. Most of time, an agent theory is expressed by modal logic which is a good specification tool since it eases the description of intentional agents. However, this formalism cannot be easily refined into implementation even if some counter-examples exist (see [3], and [4] for example). Actually a large and ugly chasm

still separates the world of formal theory and infrastructure from the world of practical nuts-and-bolts agent system development.

This paper proposes another type of formalism based on Petri nets [5] and objects. This formalism called cooperative objects[6], belongs to the class of concurrent object-oriented languages but it can also be used with benefits to both specify and implement autonomous and concurrent agents. On the one hand, cooperative objects can be considered as a specification language since they allow one to model the behavior of each agent and check some good properties concerning the behavior of the agents and of the whole system. On the other hand, this formalism can be used as an agent language enabling the implementation of a multi-agent system as concurrent agents.

The presented study shows how cooperative objects suit the needs of multi-agent systems. We mainly show how the proposed formalism favor autonomy property to a great extent. Autonomy has often been promoted as an essential, defining property of agenthood. The essence of autonomic computing systems is self-management for which four aspects are cited: self-configuration, self-optimization, self-healing, and self-protection. In this study we don't address those aspects separately. They will be emergent properties of a general architecture, and distinctions will blur into a more general notion of autonomy of maintenance.

This paper is organized as follows. Section 2 describes the main features of cooperative objects formalism. Section 3 uses the prey/predators problem as a case study and shows how to specify and implement preys and predators as cooperative objects.

2 Cooperative Objects

Cooperative objects integrate concepts from object oriented approach and from Petri nets. This formalism may be characterized by the following equations:

$$\begin{aligned} \text{System} &= \text{Objects} + \text{Cooperation} \\ \text{Object} &= \text{data structure} + \text{Operations} + \text{Behavior} \end{aligned}$$

The object behavior describes its control structure and the cooperation component defines how objects interact, according to a client /server protocol, in order to achieve the goal of the whole system. Each cooperative object is an instance of its class which determines its structure. The data structure of a cooperative object is a set of attributes which can be public (i.e., their value may be read by any other object) or private. The operations component of a cooperative object is made up with operations and services. Operations are methods in the usual meaning of object oriented languages, and allow synchronous communications (i.e., calling an operation blocks the client, whereas the server is assumed to provide a result immediately). Some of these operations are private, while the other ones are public. Services support asynchronous communication (i.e., the client is not blocked by calling a service, and the server may need some time to provide a result because it is not able to process the request, or because this processing requires a lot of work). All services are public. The behavior of an object and its cooperation with others are described by an OBCS (for Object Control Structure).

An OBCS is a Petri net with objects [7], an extension of Petri nets in which tokens are data structures. A Petri net with objects is made up with places, transitions and arcs annotated with inscriptions describing how tokens are processed. Places are the state variables of the modeled system. Each place may contain zero, one or any number of tokens of a given type. A token is either a constant value (e.g., integer, boolean, ...), an object or a reference toward an object, or a tuple of those items). Transitions aim at changing the net state, that is the tokens values and locations. Variables labeling arcs act as formal parameters of transitions. They allow to state on which tokens the transition action is applied, and they define the flow of tokens from input to output places. A transition may be guarded by a precondition which is a boolean expression testing the values of tokens bound to input variables. Each transition has a priority level. Hence a transition may occur only if its input tokens satisfy the precondition and no other transition having a higher priority is enabled. A transition may be associated with a list of boolean expressions called emission rules. Each output arc is then bound to one emission rule. At the end of occurrence of the transition, the emission rules are evaluated; one is chosen among the true ones, and only arcs associated to this emission rule are activated. A transition may occur if its input places contain tokens to which its input variables may be bound, in such a way that its precondition is true. The occurrence of a transition changes the marking of its input and output places. Tokens are removed from each input place according to arcs variables. Then the action of the transition takes place. This action is either a service call or any piece of code. The transition firings complete by putting tokens into output places according to arc variables.

Cooperative objects support the multi-tasking inside objects and asynchronous communications (in fact, these two features are tightly associated since it would make no sense to support asynchronous communications if an object would not have the possibility to do something else while it is waiting for the result of a communication). Accordingly, a cooperative object enjoys from a high level of autonomy. For instance, it can have a spontaneous activity aiming at reaching its own goal while it processes any number of requests for its services. Similarly, the autonomic nervous system carries out some functions (e.g., checking blood sugar level, adjusting pupils to the right amount of light, lowering heart rate at rest, digesting lunch, ...) across a wide range of external conditions, always maintaining a steady internal state called homeostasis while readying the body for the task at hand.

Syntactically speaking, the definition of a cooperative object class is made up with a specification and an implementation. The specification corresponds to its interface (i.e., public attributes and operations together with the declaration of its services). The implementation of a class includes the definition of its private attributes and operations, and its OBCS.

3 A Case Study

A cooperative object implementation of the Prey/Predators game requires the definition of the classes *Prey* and *Predator* together with a main procedure which initializes

the system. This procedure will create one instance of the *Prey*'s class and four instances of the *Predator*'s class, and put them on an arbitrary position upon the game field. Figure 1 shows the definition of the class *Prey*. Its interface includes two public operations : *Captive* and *GivePos*. The operation *Captive* returns true when the prey is encircled by the predators. The operation *GivePos* returns the position of the prey which may be :

- An out-position if the prey is out of the game field.
- The actual prey's position if it is in the perception scope of the caller (given as input parameter).
- A hidden position if the position of the caller is far from the prey's position.

```
Class Prey specification
Uses Position, Boolean ;
Operations
  GivePos (pclient : Position) : Position;
  Captive () : Boolean ;
End.
Class Prey implementation
Refers Predator ;
Attributes
  mypos ; //position of the prey
  vp : array [4] of Predator* ;
Operations
  move () ; // moves the prey on the grid randomly
OBCS
```

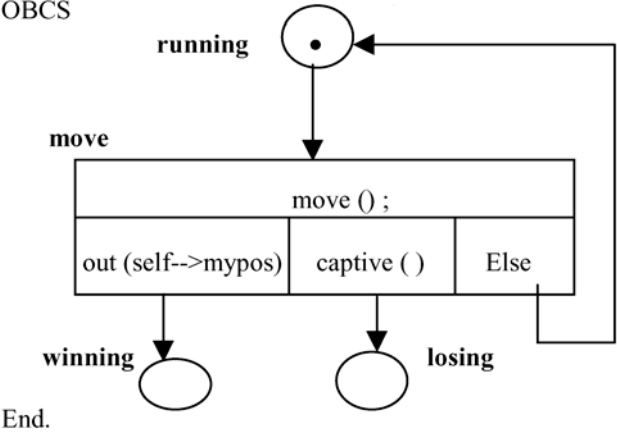


Fig. 1. Definition of the class *Prey*

The class *Prey* has two private attributes : *mypos* (the actual position of the prey on the grid), and *vp* (an array of references toward the prey's hunters).

The OBCS describes the prey's behavior. Initially, the *running* place contains one token, and the *move* transition is enabled. When it occurs, the *move()* operation is performed resulting in one of the following cases: the prey is out of the grid (then it

wins), it is captive (then it loses), or it is neither out nor captive (then it continues to move).

Figure 2 shows the definition of the class predator. The service *WherePrey* is the single item of its interface. When this service is called, the predator provides an answer only when it knows the prey's position otherwise the request is delayed until the predator knows that position. The predator's implementation is made up with :

- Four private attributes: *mypos* (the actual position of the predator on the grid), *vp* (an array of references toward the predator's colleagues with which it cooperates to capture the prey), *pr* (a reference toward the prey), and *role* (the role which must be fulfilled by the predator). This role may be : getting closer to the prey from the south, from the north, from the east, or from the west. Actually, agents can change their role dynamically at run time. This fact can be implemented by a centralized solution (it is not detailed here) such as a referee agent taking into account the respective positions of the predators as well as their current roles and gives as an output the new roles to fulfill¹.
- Two private operations: *move_r* (moves the predator randomly on the grid), and *move_t* (moves the predator toward the prey according to the current situation). With *move_t*, four scenarios should be implemented according to the current role of the predator: "attack from the south" , "attack from the north", "attack from the east", "attack from the west" .

```

Class Predator specification
Uses Position ;
Services
  WherePrey ( ) : Position ;
End.
Class Predator implementation
Uses Target;
Refers Prey;
Attributes
  mypos : Position ;
  role : Target;
  vp : array [3] of Predator* ;
  pr : Prey*;
Operations
  move_r ( ) ; // moves the predator randomly
  move_t (p:Position); //moves the predator toward the prey
OBCS // see figure 3
End.
```

Fig. 2. Definition of the class Predator

¹ A distributed solution can also be envisaged. We think, it relies on negotiation between predators, and result in the adoption of new roles. The comparison between the centralized and the distributed solutions is out of the scope of this paper.

The behavior of a predator is described by its OBCS (see figure 3). The transitions of this OBCS have no precondition, and transitions *ignore*, *move_to1* and *WherePrey* have priority 2 : they have a higher priority level than the other transitions. The type of places is written in italic characters, along their names. Initially, the predator doesn't know the prey's position. This fact is represented by the presence of one token in the *hiddenprey* place. Thus, the search transition may occur. When it does, one token is put into the *own_s* (denoting own search) place, and a reference toward the three other predators is put into the *colleagues* place.

Hence, the predator uses two means to get the prey's position:

- By its colleagues: each of the three occurrences of the *ask* transition requests the *WherePrey* service of a predator, resulting in a token into the *preyhere* place. Then, either the *move_to1* or the *ignore* transition occurs. If there is a token in the *own_s* place, *move_to1* occurs, the predator moves toward the prey, and one token is put into the *preyposition* place. If there is a token in the *preyposition* place, the *ignore* transition occurs: this enables to give up a token in the *preyhere* place, which arrived too late, at a moment where the predator already knows the prey's position.
- By its own means: when there is a token in the *own_s* place, the *move_r* transition occurs: the predator moves randomly, and calls the *GivePos()* operation of the prey, resulting in one of the following cases :
 - The prey is hidden, and then the predator carries on with moving randomly on the grid.
 - The prey has gone out of the grid and hence the predator loses.
 - The prey is within the perception scope of the predator. In this case, the prey's position is put into the *seeingprey* place.

As long as the prey stays in its perception scope, the predator pursues the prey (which is denoted in the Petri net by the repetition of the *move_to2* and *continue* transitions). Predators' cooperation to capture the prey, is analogous to the collaboration situation of autonomic managers within an IT system[8].

Finally, let's address the design/implementation distinction. In fact, cooperative objects may be used for both -they are a modeling formalism and also a programming language-, because they enable to model a system at any abstraction level, and any (correct) cooperative objects system may be executed. Thus, a cooperative objects system may be viewed either as an abstract model of a system, describing its structure, its functionalities, and its behavior, or as a program performing some functions.

A C++ implementation of cooperative objects already exists [9]. Such implementation offers facilities both for the simulation and for an efficient implementation: it compiles any cooperative object class into a C++ class able to execute its Petri net with objects, in such a way that the resulting C++ system obeys the formal semantics of cooperative objects. In the case of our example, it is enough to complete the definitions of figure 1 and 2 with the code of functions *GivePos*, *Captive*, *out* and *hidden*, and also to write the code of the main procedure, in order to play the prey/predators game.

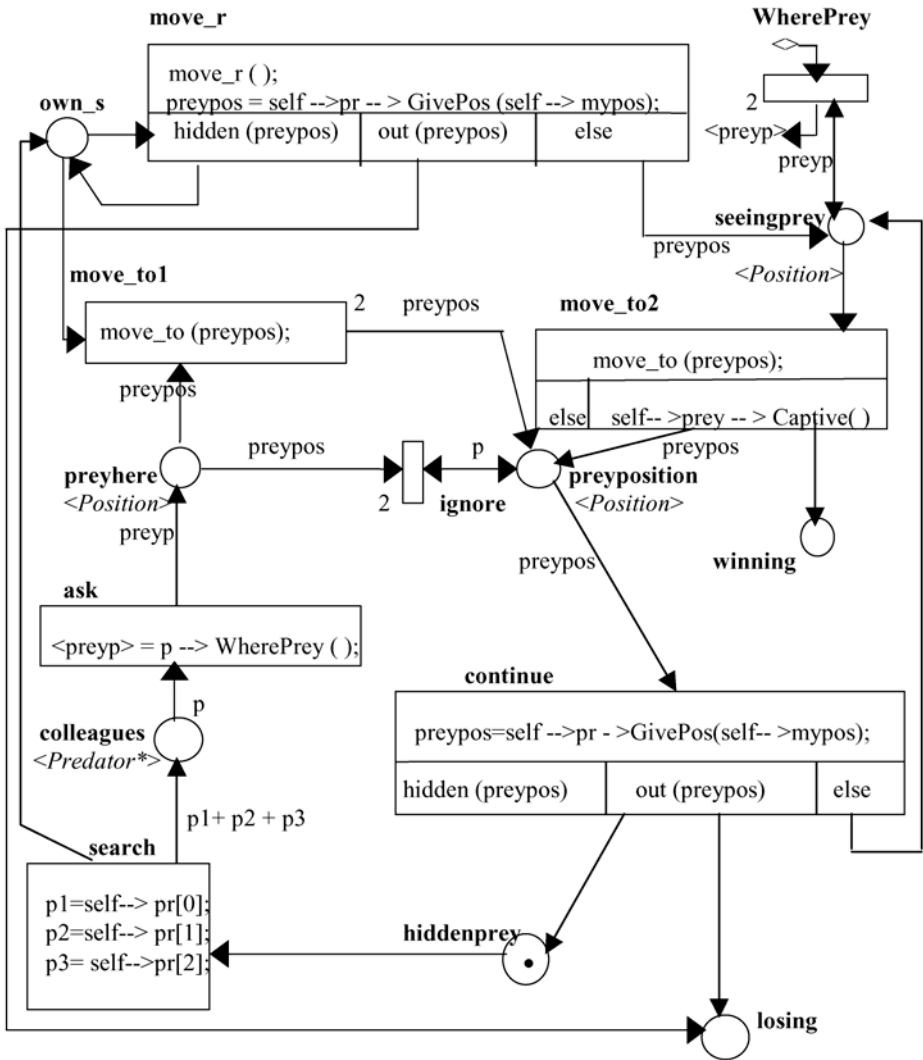


Fig. 3. The predator's control structure

4 Conclusion

Seeing agent computing as a solution to the automation of autonomic systems requires the use of agent programming languages and adequate software tools. Unfortunately, agent technology lacks mature languages which can be used on a large scale. In this paper, we have shown how cooperative objects may be used for the design, the validation and the implementation of agent systems and subsequently for the automation of autonomic systems. Indeed, an autonomic system can be viewed as a multi-

agent system whether its constituents (autonomic elements) are viewed as agents. Cooperative objects support the principles of the object paradigm which are no longer to be praised, and multitasking as well as asynchronous communications, and thus it provides objects with a high level of autonomy. The behavior of each object and its cooperation with others are defined by Petri nets with objects. The development of elements that rarely fail is an important aspect of being autonomic. Accordingly, a major challenge is to develop analysis tools and techniques for ensuring that autonomic elements will behave as we expect them to -or at least, will not behave in ways that are undesirable. With cooperative objects, the property analysis facilities are provided by the Petri nets theory.

The presented study is an endeavour to find an answer to the automation of autonomic systems via agent based computing, but there still some questions that should be addressed in the future. Some of them concern the presented Petri net formalism such as: to what extent the modification of the tokens nature can alter the analysis power of Petri nets ? Other questions deal with the essence of autonomic computing itself among which : are there any general abstractions for understanding the properties of self-configuration, self-optimization, self-healing, and self-protection ? how to express those abstractions in the Petri net based formalism, in order to use its underlying theory to prove that the modelled system satisfy the afore mentioned properties ?

References

1. J. O. Kephart, Chess, D.M : The Vision of Autonomic Computing, IEEE Computer, 41-50, 2003.
2. N. Jennings, K. Sycara, M. Wooldridge : A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems, 1, 7-38, 1998.
3. M. Wooldridge :This is MYWORLD: The Logic of An Agent-Oriented DAI Testbed. LNAI 890, Springer Verlag, 1994.
4. W. Chainbi : A Belief-Goal-Role Logic for a Single-Agent System. ACM symposium on Artificial Intelligence and Applied Computing, ISBN 1-58113-287-5, Las Vegas , ACM Press, New York, NY, USA, March 11-14, 2001.
5. T. Murata : Petri Nets: Properties, Analysis and Applications. Proc. IEEE, vol. 77, no 4, April, 1989.
6. R. Bastide : Cooperative Objects : a formalism for modeling concurrent systems. Thesis of the university Toulouse III, February, 1992.
7. C. Sibertin-Blanc : High-level Petri Nets with Data Structures. 6th European Workshop on Applications and Theory of Petri Nets, Espoo, Finland, June 1985.
8. IBM: An architectural blueprint for autonomic computing. April 2003.
9. C. Sibertin-Blanc, Manuel de référence de SYROCCO version 6, Copyright SEPT, October, 1995.

A Coverage-Preserving Node Scheduling Algorithm for Self-organized Wireless Sensor Networks

Jie Jiang and Wenhua Dou

School of Computer Science
National University of Defense Technology
410073, Changsha, China
jiangjie@nudt.edu.cn

Abstract. Self-organized wireless sensor networks are ad-hoc networks containing a large number of energy-constrained sensors. Reducing energy consumption and prolonging network lifetime is one of the most important design challenges in wireless sensor network. In this paper, a decentralized node scheduling algorithm is proposed to keep a minimal necessary number of sensors active while maintaining networking connectivity and sensing coverage. Experimental results show that our algorithm outperforms the PEAS algorithm and the sponsor area based algorithm with respect to the number of working sensors needed.

1 Introduction

Wireless sensor network (WSN) is an ad hoc multi-hop network containing a large number of resource-constrained sensors, which are capable of sensing, processing and communicating among each other using wireless radio. WSN has many potential applications, such as battlefield surveillance, target detection and localization, industry control and monitoring, and many others [5, 6].

In WSN, sensors are usually deployed in the region of interest with large number and high density (up to 20 *nodes/m*³ [1]). Sensing coverage redundancy will inevitably occur. The redundant sensing data, the corresponding wireless communication collision and interference will cause much energy to be wasted. So it is desirable that only a subset of sensors are kept working (active) while these active sensors can maintain the sensing coverage and communication connectivity. An effective method to realize above object is node scheduling, which keeps part of sensors in active mode while sensing coverage and communication connectivity are maintained.

In this paper, the issue of scheduling sensors' activity is addressed. Network lifetime is divided into round and in each round, we try to find a minimum subset of sensor nodes while these nodes can preserve the sensing coverage and maintain communication connectivity. A novel approach to judge sensing redundancy is proposed, and based on this approach, an effective, distributed and localized node scheduling algorithm is proposed for sufficiently densely deployed wireless

sensor network. Firstly, by keeping a minimum subset of nodes working while still preserving the original sensing coverage, our algorithm can reduce system energy consumption without sacrificing any quality of service provided by sensor network; secondly, when some active nodes fail, the non-active, redundant nodes can resume working to replace those nodes and keep network functioning. By this means, the overall network energy consumption can be reduced and lifetime can be prolonged. Experimental results show that our algorithm outperforms the PEAS algorithm [2] and the sponsor area based algorithm [3].

The rest of this paper is organized as follows. Section 2 reviews the related work in the literature. Section 3 describes the proposed node scheduling algorithm in detail. Section 4 presents the experimental results and section 5 concludes the paper.

2 Related Work

How to prolong system lifetime by reducing node's energy consumption is an important research challenge for wireless sensor networks. Many research efforts have been dedicated to this field.

In paper [7], Slijepcevic *et al.* consider the problem of dividing sensors into disjoint sets while each node set can provide the complete coverage of the monitored area. By activating node set in turn, network lifetime can be prolonged. The problem of finding maximum number of disjoint node sets is NP-complete. A centralized solution to this problem is proposed in [7].

In paper [2], Ye *et al.* propose a distributed, probing based density control mechanism for robust sensing coverage (PEAS). Different coverage redundancy can be achieved by adjusting node's probing range. PEAS cannot completely preserve the original sensing coverage after turning off some nodes.

In paper [3], Tian *et al.* propose a distributed and localized coverage-preserving node scheduling scheme. Although this scheduling scheme can preserve the original sensing coverage, but it has two flaws: (1) The area of sponsor sector is always smaller than the area of the crescent intersection, some overlapped area is not considered in the coverage calculation; (2) It only considers neighboring sensors within sensing range R_s . Nodes far away than R_s , but within $2R_s$ (equal to the radio range R_c) are ignored. In fact, those one-hop, direct neighboring sensors can also do help in reducing the number of working nodes needed (as shown in Fig.1). The performance of the sponsor area based node scheduling algorithm can be further improved.

In paper [4], Zhang *et al.* present a decentralized and localized density control algorithm called OGDC. OGDC gives rules (R1–R4) that specify what action one node should adapt and how to change state. But OGDC assumes that the sensor density is so high that a sensor can be found at any desirable point. It seems not realistic in practice. And if no sensor can be found to cover the crossing point, the coverage performance of the algorithm is questionable.

3 Decentralized and Localized Node Scheduling

3.1 System Model and Assumptions

In two-dimension plane, all sensors have the same sensing range (R_s) and communication range (R_c). And the sensing area of a sensor is modelled as a circle, whose center is the sensor and radius is the sensing range. We use binary sensing model, that is to say, a point p can be covered by sensor s_i if and only if the Euclid distance between p and s_i , $d(p, s_i) < R_s$. Area A is covered by S if and only if every point in A is covered by at least one sensor in set $S = \{s_1, s_2, \dots, s_n\}$. All sensors in the network know their own positions and are time-synchronized. Many research efforts dedicated to localization problem [8] and time synchronization problem [10] in wireless sensor network make this assumption realistic.

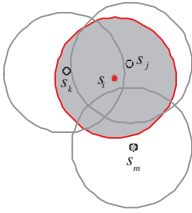


Fig. 1. Scenario ignored by node scheduling algorithm in [3]

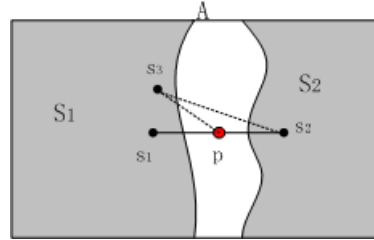


Fig. 2. Proof of Theorem 1

3.2 Localized Coverage Redundancy Detection

To keep a minimum subset of sensors active, there must be a rule for sensor node to decide whether it is safe to be non-active, that is, even it is turned off, the whole sensing coverage can still be maintained and the detection/monitoring performance is not reduced. Although some applications require all points in the monitored field to be covered by several sensors to improve the reliability and accuracy, we focus on those applications which require each point to be covered by at least one sensor and try to maximize the whole network lifetime. It is obvious that if a sensor's sensing area is covered by its neighboring sensors, then turning it off will not cause any sensing hole and the original coverage of the sensor network can be maintained. At the same time, each node must detect whether it is redundant in coverage in a distributed and localized manner (just utilizing the information from local neighbor sensors). In this subsection, we give some definitions and theorems that form the base of our node scheduling algorithm.

Definition 1 (Neighbor). If the Euclid distance between sensor s_i and s_j satisfies $0 < d(s_i, s_j) \leq R_c$, where R_c is sensor's radio radius, then s_i and s_j are neighbors to each other. The set of s_i 's neighbors is denoted as $N(i) = \{s_j | 0 <$

$d(s_i, s_j) \leq R_c, s_j \in S\}$, where $S = \{s_1, s_2 \dots s_n\}$ is the set of all sensors deployed in the region of interest.

In multi-hop WSN, to ensure every sensor's data can reach the sink (base station), each active sensor must be connected to the sink through neighbors. Therefore, when designing the node scheduling algorithm, we must take into consideration both the sensing coverage and communication connectivity. It is required that after scheduling, the active sensors must be connected to the sink as well as maintain the original sensing coverage. Here we first explore the relationship between sensing coverage and communication connectivity.

Theorem 1. *Assume the region of interest (A) is finite and convex. If the sensor's radio radius is at least twice of the sensing radius, i.e. $R_c \geq 2R_s$, then the region A is completely covered by S implies that all sensors in S are connected.*

Proof. This theorem can be proved by contradiction. Suppose $R_c \geq 2R_s$ and the region A is completely covered by S . Assume S is not connected. That is, S is at least divided into two disconnected subset S_1 and S_2 , where $S_1 + S_2 = S$ (see Fig.2). Here S_1 is the set of sensors located in the left shaded area of A and S_2 is the set of sensors located in the right shaded area of A . Note that S_1 and S_2 are connected individually. We use $d(S_1, S_2) = \min d(s_i, s_j)$ (where $s_i \in S_1, s_j \in S_2$), to denote the distance between S_1 and S_2 . Since S_1 and S_2 is disconnected, then $d(S_1, S_2) > R_c$.

Assume $s_1 \in S_1, s_2 \in S_2$, and $d(s_1, s_2) = d(S_1, S_2)$. We draw a line between s_1 and s_2 . Consider the midpoint p of the line segment $\overline{s_1 s_2}$. Since A is completely covered by S , point p must be covered by at least one sensor. Not losing generality, assume p is covered by sensors in S_1 . If p is covered by s_1 , then $d(p, s_1) < R_s$ and $d(s_1, s_2) = 2d(p, s_1) < 2R_s$. If p is covered by another sensor, say $s_3 \in S_1$, but not covered by s_1 . Then $d(p, s_3) < R_s$ and $d(p, s_1) \geq R_s$. So $d(p, s_3) < d(p, s_1)$. From triangular inequality, $d(s_3, s_2) \leq d(p, s_2) + d(p, s_3) < d(p, s_2) + d(p, s_1)$. That is, $d(s_3, s_2) < d(s_1, s_2)$. This is contradictory to $d(s_1, s_2) = d(S_1, S_2)$. So if point p is at least covered by one sensor in S_1 , it must at least be covered by s_1 . Then $d(p, s_1) < R_s$, and $d(s_1, s_2) < 2R_s$.

From above discussion, $d(S_1, S_2) > R_c$ and $d(S_1, S_2) = d(s_1, s_2) < 2R_s$, we get $R_c < 2R_s$. This is contradictory to the precondition ($R_c \geq 2R_s$). So set S is connected. $\square$

Theorem 1 establishes a sufficient condition for a completely covered network to guarantee connectivity. Under the condition that $R_c \geq 2R_s$, we can integrate sensing coverage and communication connectivity into one framework and focus only on the sensing coverage maintenance. As longer R_c is, more interference occurs, we set $R_c = 2R_s$ in the following description.

Definition 2 (Perimeter coverage [9]). *If a point on the perimeter of the sensor s_i 's sensing circle is covered by sensor s_j , we say the point is perimeter covered by s_j . If every point on the perimeter of s_i 's sensing circle is perimeter covered by at least one neighbor, the whole sensing circle is perimeter covered by neighbors. Similarly, if every point on a segment of s_i 's sensing circle is perimeter covered by at least one neighbor, then the segment is perimeter covered.*

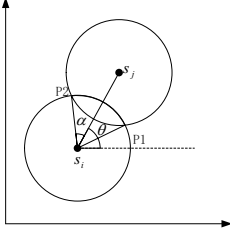


Fig. 3. Calculation of central angle

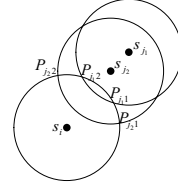


Fig. 4. Effective neighbor

Consider sensors s_i and s_j , located at (x_i, y_i) and (x_j, y_j) individually as shown in Fig.3. The range of the central angle corresponding to the s_i 's segment $\widehat{P_1P_2}$ (along the counterclockwise direction) covered by s_j is denoted as $\theta_{j \rightarrow i} \triangleq [\theta - \alpha, \theta + \alpha]$, where $\theta = \arctan \frac{y_j - y_i}{x_j - x_i}$ and $\alpha = \arccos \frac{d(s_i, s_j)}{2R_s}$.

Theorem 2. If $\bigcup_{j \in N(i)} \theta_{j \rightarrow i} \neq [0, 2\pi]$, s_i 's sensing area is not completely covered by neighbors.

The proof of this theorem is omitted because it can be easily proved by contradiction.

Theorem 2 gives a sufficient condition for s_i to decide whether it is eligible for off-duty. This local decision making is fast and can accelerate the decision process of nodes non-eligible for off-duty.

Note that one sensor may have many neighbors in high-density environment. By introducing the concept of “effective neighbor”, we show that it is sufficient to consider only the effective neighbors to decide whether it is redundant in sensing coverage.

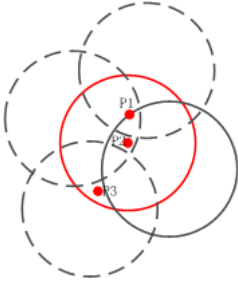
Definition 3 (Effective neighbor). Suppose s_{j_1} and s_{j_2} are s_i 's neighbors (see Fig.4). If $\theta_{j_1 \rightarrow i} \subseteq \theta_{j_2 \rightarrow i}$, we say s_{j_2} is s_i 's effective neighbor and s_{j_1} is not.

Since $\theta_{j_1 \rightarrow i} \subseteq \theta_{j_2 \rightarrow i}$, $\bigcup_{j \in N(i)} \theta_{j \rightarrow i} \subseteq \bigcup_{j \in N'(i)} \theta_{j \rightarrow i}$, where $N'(i)$ is the set of s_i 's effective neighbors. In the following discussion, we only consider effective neighbors.

Theorem 3. Sensor s_i 's sensing area is at least 1-covered (without considering s_i), iff the arc segment of each effective neighbors' sensing perimeter, lying within s_i 's sensing circle, is perimeter covered by s_i 's other effective neighbors.

Proof. Consider sensor s_i and its effective neighbor set $N'(i)$. (From now on, we only consider effective neighbors.) s_i 's sensing area (bounded by the red circle) is divided into many subregions by neighbors' sensing boundary (as shown in Fig.5).

The “only if part” is obvious since all points within s_i 's sensing area are covered by neighbors.

**Fig. 5.** Proof of theorem 3

```

Procedure is_coverage_redundant ( $s_i$ )
begin
  1. Construct effective neighbor set  $N'(i)$ 
  2. If  $\bigcup_{j \in N'(i)} \theta_{j \rightarrow i} \neq [0, 2\pi]$ 
    return FALSE
  3. If the “if part” of theorem 3 holds
    return TRUE
  else
    return FALSE
end

```

Fig. 6. Pseudo code for coverage redundancy detection algorithm

Now we show the “if part”.

For any point p within s_i ’s sensing area, p must be on the arc segment of some sensor’s sensing boundary or in one of those subregions.

Case(i): p is on the arc segment of some sensor’s sensing boundary (see p_1). According to the precondition that each effective neighbor’s sensing boundary within s_i ’s sensing area is perimeter covered, then p is also covered by other neighbors.

Case (ii): p is within one subregion. There are two types of subregions. One type is that the region is formed by at least one neighbor’s interior arc segment; the other is that the region is only formed by neighbors’ exterior arc segment and s_i ’s sensing boundary. For the former case, p is covered by the owners of the interior arc segments (see p_2). Its coverage degree is at least 1 (without considering sensor s_i). For the latter case (see p_3), p ’s coverage degree is equal to that of points on the exterior arc segment. Since every neighbor’s arc segment lying within s_i ’s sensing circle is perimeter covered, the coverage degree of points on these arc segments is at least 1, hence p is also at least 1-covered.

Since p is at least 1-covered without s_i , considering the arbitrariness of p , we show that s_i ’s sensing area is at least 1-covered by its effective neighbors. $\square$

Knowing the position information of each neighbor, sensor s_i can locally decide whether one neighbor’s arc segment within its sensing area is perimeter covered by other neighbors or not. Because of the page limit, here we don’t outline the formula in detail.

Based on the above discussion, we describe our localized coverage redundancy detection algorithm in Fig.6.

3.3 Distributed Node Scheduling

In this subsection, we propose a distributed and localized node scheduling algorithm to reduce the number of working sensor nodes needed in Fig. 7.

1. *Upon entering the Neighbor Discovery phase:*
 - (a) set *timer* to be *discovery_interval* (T_d)
entering *Evaluating phase* upon timeout
 - (b) broadcast *hello* message to one-hop neighbors after a random time slot
2. *Upon entering the Evaluating phase:*
 - (a) set *timer* to be random *wait_interval* (T_w)
wait until timeout
 - (b) **If** no *OFF* message has been received
 - (b.1) call *is_coverage_redundant* (s_i)
 - (b.2) **If** *is_coverage_redundant* (s_i)=FALSE
Keep *ACTIVE* state
Done and stop
 - (b.3) **If** *is_coverage_redundant* (s_i)=TRUE
Broadcast *OFF* message to one-hop neighbors
Turn off communication and sensing units
change to *NON-ACTIVE* state
Done and stop
 - (c) **If** one or more *OFF* message has been received
Delete the source node of the *OFF* message from neighbor list
goto (b.1)

Fig. 7. The activity scheduling protocol at sensor s_i

In our algorithm, a sensor is in one of the two states: “ACTIVE” and “NON-ACTIVE”. At the beginning, all nodes are in ACTIVE state. Network lifetime is divided into rounds, and each round has a scheduling phase followed by a sensing phase (see Fig.8). The scheduling phase is further divided into two sub-phases: neighbor discovery phase and evaluating phase. To minimizing the energy overhead consumed in the scheduling phase, the sensing phase should be long enough compared to the scheduling phase. In each time round, the ACTIVE nodes work for the sensing task and the NON-ACTIVE nodes turn off their sensing and communication units to save energy.

4 Experimental Results

In this section, we present some experimental results as the performance evaluation of our algorithm. We compare our algorithm with PEAS because PEAS can maintain approximately the original sensing coverage (more than 99%) when the probing rang is short enough. And since the sponsor area based node scheduling algorithm is the base of our work, we will focus on the performance comparison with it. To show the effectiveness of our algorithm in energy efficiency, we also compare the average sensing degree before and after turning off some nodes.

4.1 Comparison with PEAS

To compare our algorithm with PEAS, we carry some experiments in static networks. In a square field ($50m \times 50m$), we deploy 100 sensor nodes randomly.

Sensor uses binary sensing model, sensing radius is $10m$ and radio radius is $20m$. We consider the performance of PEAS under different probing range and the results in Table 1 are obtained as the average of 100 random topologies respectively. Note the coverage degree of a point is defined as the number of sensors that can cover this point. Sensing hole occurs when a point is covered by original network but not covered any more after some nodes are turned off. To calculate the coverage degree and sensing holes, we divide the deployment field into $1m \times 1m$ grids, and only consider the center of these grids.

From Table 1, we can see that PEAS can obtain approximately the same off-duty (NON-ACTIVE) node number as our algorithm only when the probing range is longer than $6m$. But there are 36 sensing holes in that case.

Table 1. Performance Comparison with PEAS

Algorithm	Probing range	# of off-duty nodes	Original sensing degree	Obtained sensing degree	# of topologies with sensing holes	Average # of sensing holes per topology
Proposed	N/A	75	10	3	0	N/A
PEAS	3	36	10	6	14	< 1
	4	50	10	4	24	2
	5	63	10	3	63	7
	6	72	10	2	92	36
	7	80	10	1	100	103

4.2 Comparison with Sponsor Area Based Scheduling Algorithm

To our best knowledge, the sponsor area based node scheduling algorithm is the only scheme that can preserve the original sensing coverage after turning off some nodes. From theorem 3, a sensor can only be turned off only if its sensing area is covered by its neighbors. Hence our scheduling algorithm can also preserve 100% sensing coverage of the original network. But our algorithm can obtain more off-duty eligible nodes than the sponsor area based scheduling algorithm, thus longer system lifetime can be expected.

We use the same setup as the previous experiment. Fig.9 shows the obtained non-active node number with different sensing range and different deployed node number. We can see that our algorithm can obtain about 30% more non-active nodes compared to the sponsor area based scheduling algorithm under different sensing range. The performance improvement is obtained by extending the range of neighbors (the distance between node and its neighbors is extended from R_s to $2R_s$) and the new method for coverage calculation.

Fig.10 is a 3-D surface plot of working node number in different deployment density. We can see from it that the number of working nodes needed to preserve the original sensing coverage is much smaller than that of the sponsor area based algorithm. Experimental result shows that our algorithm can effectively control the working node number. When original deployed node number increases from 100 to 300, the number of working nodes increases only about 20%.

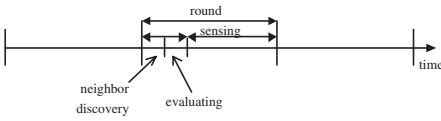


Fig. 8. Network lifetime

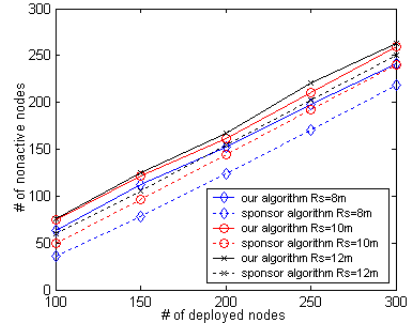


Fig. 9. # of nonactive nodes vs. node density

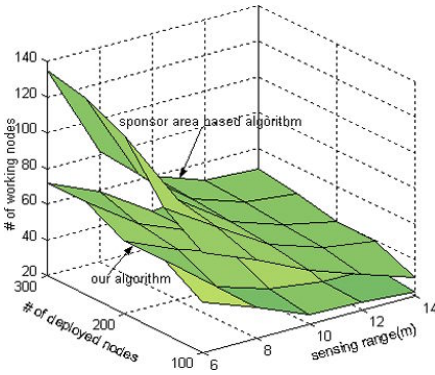


Fig. 10. # of active nodes vs. node density

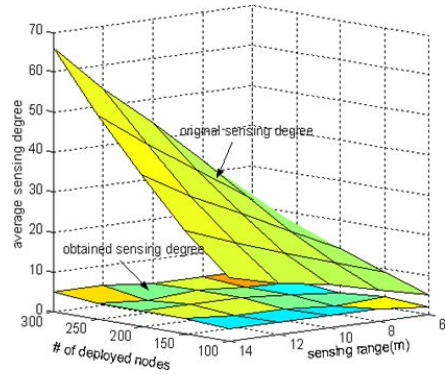


Fig. 11. Sensing degree vs. node density

4.3 Average Sensing Degree vs. Node Density

Another metric that can prove the effectiveness of our algorithm in energy saving is the resulted average sensing degree of the deployed area. Sensing degree can reflect the sensing redundancy of the network. Higher sensing degree will result in more redundant data, more traffic load and more wireless communication collision, which will waste more energy. As illustrated in Fig.11, although the original sensing degree varies from 5 to 66, our scheduling algorithm can result in about 3 degree.

5 Conclusions and Further Work

In this paper, we present a coverage-preserving, distributed node scheduling algorithm for wireless sensor networks. The algorithm can reduce overall system energy consumption, therefore increase network system lifetime, by turning off

some redundant nodes. Experimental results show that our algorithm outperforms the PEAS algorithm and the sponsor area based algorithm. Current algorithm needs node's position information thus will rely on localization service, which is not very easy and cheap in sensor networks. Our next work is to develop new algorithm that will not depend on geographical location information. The scheduling problem of WSN in 3D space is our another direction.

References

1. Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A.: Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In Proc. of MobiCom' 01, Rome, Italy, July, 2001.
2. Ye, F., Zhong, G., Lu, S., Zhang, L.: Peas: A robust energy conserving protocol for long-lived sensor networks. In Proc. of the 23rd ICDCS, 2003.
3. Tian, D., Georganas, N.D.: A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Network. In Proc. of WSNA'02, Atlanta, Georgia, USA, 2002.
4. Zhang, H., Hou, J.C.: Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Technical Report. UIUCDCS-R-200302351. UIUC. June, 2003.
5. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks. March, 2002.
6. Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: Mobile networking for Smart dust? In Proc. of of MobiCom'99, Washington, August, 1999.
7. Slijepcevic, S., Potkonjak, M.: Power Efficient Organization of Wireless Sensor Networks. In Proc. of ICC'01, Helsinki, Finland, June, 2001.
8. Nasipuri, A., Li, K.: A Directionality based Location Discovery Scheme for Wireless Sensor Network. In Proc. of WSNA'02, Atlanta, Georgia, USA, September 28, 2002.
9. Huang, C-F., Tseng Y-C.: The Coverage Problem in a Wireless Sensor Network. In Proc. of WSNA'03, San Diego, California, USA, September 19, 2003.
10. Elson, J., Romer, K.: Wireless Sensor Networks: A New Regime for Time Synchronization. ACM Mobile Computing and Communication Review (MC2R), Vol.6 No.4, pp.59-61. October, 2002.

Evaluation Issues in Autonomic Computing

Julie A. McCann and Markus C. Huebscher

Imperial College London, UK
{jamm,mch1}@doc.ic.ac.uk

Abstract. Autonomic computing is a concept that brings together many fields of computing with the purpose of creating computing systems that are reflective and self-adaptive. In this paper we draw upon our experience of this field to discuss how we can attempt to evaluate autonomic systems. By looking at the diverse systems that describe themselves as autonomic, we provide an introduction to the concepts of autonomic computing and describe some achievements that have already been made. We then discuss this work in terms of what is necessary to evaluate and compare such systems. We conclude with a set of metrics, which we believe are useful to evaluate autonomicity.

1 Introduction

Autonomic computing is generally considered to be a term first used by IBM in 2001 to describe computing systems that are said to be self-managing [11]. However the concept of self-management and adaptation in computing systems has been around for some time. The event of the combination of object-oriented programming paralleled with component-based software engineering (dynamic reconfiguration) essentially paved the way toward autonomic computing.

When reviewing the current state-of-the art in autonomic systems, the concept of self-management usually groups into four basic properties: self-configuration, self-optimization, self-healing and self-protection. Here is a brief description of these properties (for more information, see [11, 2]):

Self-configuration An autonomic computing system configures itself according to high-level goals.

Self-optimisation An autonomic computing system optimises its use of resources. It may decide to initiate a change to the system pro-actively (as opposed to reactive behaviour) in an attempt to improve performance.

Self-healing An autonomic computing system detects and diagnoses problems. What kinds of problems are detected can be interpreted broadly: they can be as low-level as a bit-error in a memory chip (hardware failure) or as high-level as an erroneous entry in a directory service (software problem) [19].

Self-protection An autonomic system protects itself from malicious attacks but also from end users who inadvertently make software changes, e.g. by deleting an important file. The system autonomously tunes itself to achieve security, privacy and data protection.

However, as there is no agreed definition of what an autonomic system is, their evaluation and moreover comparison is difficult. Furthermore the very emergent nature of such systems adds further complexity to the evaluation of such systems. This paper is an attempt to look at autonomic computing and try to highlight areas, which can be used to compare performance and derive some form of metrics.

The structure of this paper is as follows. Initially, in section 2 we survey the area of autonomic computing to attempt to build a map of the subject. To this end we provide an introduction to the concepts of autonomic computing and describe some research that is taking place in various fields of computing and some achievements that have already been made. In section 3, we concentrate on research in the field of software engineering and describe projects that focus on adding autonomic behaviour to software systems. Finally, in sections 4 and 5 we combine this work together with a discussion on performance evaluation and benchmarking, taking into account our experiences of measuring autonomic systems and provide some initial ideas on how such systems can be compared.

2 Why Autonomic Computing

In trying to understand how to evaluate an autonomic system one must understand the reason we would want such a system. This allows us to compare whether or not its objectives have been met.

The main reason for large blue-chip companies, like IBM, being interested in autonomic computing is the need to reduce the cost and complexity of owning and operating an IT infrastructure [20, 15]. In particular, there is a need to alleviate the complexity with which system administrators of IT services are faced today. The aim is to allow administrators to specify high-level policies that define the goals of the autonomic system, and let the system manage itself to accomplish these goals. At present, system administrators must tweak hundreds of settings and often spend weeks before getting a system to run optimally. Autonomic systems are also faster at adapting to changes to the environment, e.g. by distributing its resources differently when a critical project requires more CPU processing power. Furthermore, as information systems in enterprises grow larger, it is becoming increasingly difficult to identify a failure in the system and repair the affected component quickly, as large systems are heterogeneous and no single person knows the entire system.

Autonomic behaviour is a topic that has found its way in many other computing fields, in particular ad-hoc networking. For example, Liu and Martonosi [14] discuss the problem of propagating software updates in a wireless network of devices that are spread over a large area and are not all reachable from a base station. Sensors cooperate to propagate software updates to the entire network of sensors, but at the same time they must optimise energy consumption, because of tight energy constraints. Further, due to the autonomous nature of NASA's DS1 (Deep Space 1) mission and the Mars Pathfinder, some self-adaptation was required. That is, as mission control cannot rapidly send new commands to a

probe, it must quickly adapt to extraordinary situations, therefore it is important that a probe is able to make decisions and carry them out on its own.

3 Software Architectures for Autonomic Computing

The autonomic research activities in software systems can broadly be categorised into four areas: monitoring of components, interpretation of monitored data, creation of a repair plan (i.e. an adaptation of the system), and execution of a repair plan. Based on this, we choose to group the approaches to autonomic computing systems orthogonally into two categories: tightly coupled and decoupled autonomic systems. However, the two approaches have common concepts and it is sometimes difficult to place a research project in one particular category.

3.1 Tightly Coupled Autonomic Systems

Tightly coupled systems are often built using intelligent agents. Every agent has its own goals, which drive its decisions. An agent in an autonomic system is pro-active, and possesses social ability [24]. However, there are some drawbacks to this architecture. For example, the chain reaction of agents instructing other agents to change behaviour can potentially lead to instabilities of the overall system [11]. Further, a difficult task is also to define the individual goals of the agents such that the desired global goal is accomplished [11]. In an autonomic system, we want to be able to provide goals in the form of high-level notions, and expect the agents themselves to determine what behaviour is necessary to reach them.

Wise et al. [26] propose a top-down hierarchical coordination model for agent applications, in the form of their visual process language Little-JIL. A task is divided into steps, and each step can further be divided into sub steps. A step can then be assigned to an execution agent, which keeps an agenda of tasks to complete.

Although in multi-agent systems each component exhibits its own autonomic behaviour, there is usually a clean separation between the conventional component that performs a task and the autonomic manager which implements self-management around it. Figure 1 (based on a figure from [22]) shows a general logic diagram for an autonomic agent. However, in some systems the autonomic logic is tightly embedded in the main application logic of the agent.

Compared to the decoupled approach, adaptive multi-agent systems have the advantage of an innate distributed architecture (lowering the number of central

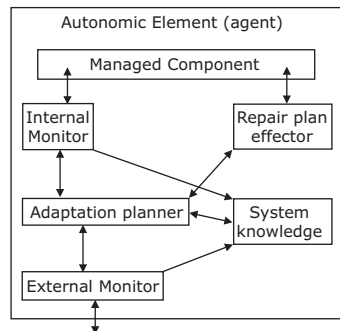


Fig. 1. Decoupled autonomic systems

points of failure). With no centralised monitoring infrastructure, agents monitor themselves (internal monitor) but also other agents (external monitor). External monitoring can be achieved pro-actively by having each agent send its heartbeat or pulse regularly on an autonomic signal channel that other agents send and listen on [22], Globus Heartbeat Monitor¹.

For example, Kumar and Cohen [13] show with experimental data how a team of broker agents can recover when a broker agent gets disconnected from the rest of the system. Again Broker agents share the same global knowledge of the system, and therefore when a broker agent discovers that another agent has been disconnected, it shares this information with the rest of the team. Also, Bigus et al. [3] are extending their ABLE agent platform to support autonomic agents to reduce the system administrator workload.

3.2 Decoupled Autonomic Systems

In the decoupled approach, the individual components are not per se autonomic. Instead, the infrastructure that handles the autonomic behaviour of the system uses an architecture description model of the running system (which is not necessarily autonomic in itself) to monitor the running system, reason about it and determine appropriate adaptive actions. The adaptivity infrastructure is typically clearly separated from the running system.

First of all, an architecture model is used to design the system (as is often the case with software development). In essence, an architecture model can be considered a graph of interacting components [7, 10, 25] and Cougaar². The nodes of a graph are called *components*, a general concept, and what a component actually is depends on the application. The arcs in the graph are called *connectors* and they represent the interaction paths between components. Many systems allow components and connectors to be annotated with a property list and constraints [7, 5, 18]. These properties are updated during monitoring of the running system and the constraints on them are used to decide when an adaptation is necessary. The autonomic infrastructure is therefore loosely coupled with the running system. In fact, it can run on a different machine, so as not to hinder the running system [7]. In some examples of such architectures, the code of components is augmented with checkpoints, e.g. to allow reporting of the occurrence of specific method calls, thereby making monitoring more straightforward.

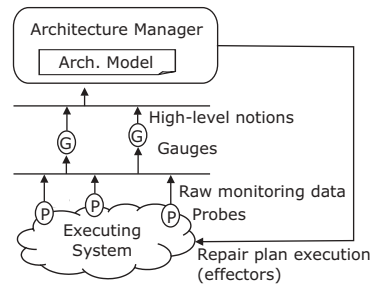


Fig. 2. An autonomic agent

¹ The Globus Heartbeat Monitor Specification v. 1.0. URL:

http://www-fp.globus.org/hbm/heartbeat_spec.html

² Cognitive Agent Architecture. URL: <http://www.cougaar.org/>

Monitoring. Figure 2 shows a diagram of a decoupled autonomic system illustrating the monitoring infrastructure (it is based on figures from [7, 23, 18]). *Probes* can be inserted into the running system to monitor it. These probes are usually localised and deliver system-specific observations [9]. The raw monitoring data provided by the probes is then aggregated and mapped to high-level notions in the architecture model by so-called *gauges*. When a property in the architecture model is updated through monitoring, the architecture model is analysed to determine whether the system is still performing adequately. If not, a repair plan is created. The repair plan is based on repair strategies that are defined in advance. That is, for many of the architectures the adaptive strategy is closed, however in the future we may see the knowledge of the success of past repair plans used to determine the best strategy [8]. This can be determined “off-line” on another machine, however a considerable amount of bandwidth may be required for monitoring, and this can become a problem if the monitoring data travels on the same network interface as application data, as experienced in Patia [18, 7].

An advantage of the complete separation between autonomic behaviour and the running system is that software adaptation can be “plugged into” a pre-existing system [23].

Hot Swapping Components. Much research has been carried out with regard to the hot swapping of components to reconfigure a system [6, 5, 16]. Typically this involves various stages: terminating a component that is to be replaced and suspending any components and connectors bordering the affected area; removing components and connectors and adding new ones as defined by the repair plan; and resuming components and connectors affected.

Rutherford et al. [21] show how an Enterprise JavaBeans system can be extended such that components can be replaced with new versions. Preliminary experiments show that loading a new component and binding it in the system takes in the order of a few seconds.

Other approaches do not require entire components to be terminated, removed and replaced [25]. Modelling at the level of code blocks allows efficient adaptation of component behaviour. Here adaptivity is fine-grained, but it permeates the design of the system down to the code blocks.

Appavoo et al. [1] show how the component-based operating system K42 has been improved to support hot swapping of components in the OS. The notion of a component here is fine-grained: a component is for example the File Cache Manager (FCM) (also see [12]).

4 Metrics and Evaluation

With modern computing – consisting of new paradigms such as planetary-wide computing, pervasive and ubiquitous computing – systems are more complex than before. Interestingly, when chip design became more complex, we employed computers to design them and today we are now at the point where humans

have limited input to chip design. With systems becoming more complex it is a natural progression to have the system not only automatically generate code but also carry out the day-to-day running and configuration of the live system. Therefore, autonomic computing has become inevitable and therefore will become more prevalent. Hence their evaluation is becoming increasingly important. This section lists a sets of metrics and means by which we can compare such systems.

4.1 Quality of Service (QoS)

QoS is possibly the top-level means to compare modern systems – it should reflect the degree to which the system is reaching its primary goal. It is typically composed of a number of metrics, e.g. data delivery time over cost. It is a highly important metric in autonomic systems as they are typically designed to improve some aspect of a service. Most of the research in this field is looking at using autonomicity to improve performance (usually speed or efficiency). However other systems wish to improve the user's experience with the system in self-adaptive or personalised GUI design for disabled people. Overall this metric is tightly coupled to the application area or service that is expected of the system. It can be measured as a global goal metric or at the sub-service or component level where each unit's ability to meet its local goal is measured.

4.2 Cost

Autonomicity costs, but the degree of this cost and its measurement is not clear-cut. Currently, most performance studies of architecture design-based autonomic systems have measured its ability to reach its goal. However more appropriately, agent-based systems typically compare the amount of communication, actions performed, and cost of actions required to reach the goal.

For many commercial systems, the aim is to improve the cost of running an infrastructure, which includes primarily people costs in terms of systems administrators and maintenance. This means that the reduction in cost for such systems cannot be measured immediately, but over time and as the system becomes more and more self-managing.

Cost comparison is further complicated by the fact that adding autonomicity means adding intelligence, monitors and adaptation mechanisms – and these features cost. In one of our autonomic computing projects, Patia, our aim was to measure the cost of adding autonomic features to a web server that can cope with fluctuating and sudden high demand (flash crowds) [18]. We found that the costs of adding monitors and monitor traffic were only just outweighed by the benefits they provided under the normal operation of the server. As this was fairly predictable it was hardly worthwhile. However, under duress the system would not work without the autonomic features. Therefore, would a comparative characteristic be an additional functionality in a system that would otherwise not be achieved in a non-autonomic system? As this might be found

in a serendipitous fashion, it could be difficult to predict what to test for in advance.

The system's architecture can also impact the measurement of the cost of a self-adaptive system. For example, most architecture design-based solutions consist of a service that has autonomic features added. For many of these architectures, the intelligence to run the system is separate and centralised, the monitors or gauges are external to what they are measuring and the decision to adapt and its supervision is external to the component. Here the question is: do we compare systems that use other computing nodes to run the autonomic services with those that run the autonomic services on the same system? With the former, costs could be in terms of extra hardware and communications to that hardware node, and the saving is that it lessens the impact on the execution of the main system. Extra nodes dedicated to the autonomic services means that they could be more intelligent, checking the validity of a given reconfiguration or provide open intelligence where the autonomic decisions themselves are adaptive. However, in agent-based autonomic systems, the intelligence is highly distributed and usually contained within the component or agent. Therefore, the self-management overhead is perhaps indistinguishable from the agent's core function and as a result it is more difficult to separate out the costs of autonomicity – if sensible at all.

4.3 Granularity/Flexibility

The granularity of autonomicity is an important issue when comparing autonomic systems. Fine-grained components with specific adaptation rules will be highly flexible and perhaps adapt to situations better, although this may cause more overhead in terms of the global system. That is, if we assume that each finer-grained component requires environmental data and is providing some form of feedback on its performance, then potentially there is more monitoring data, or at least environmental information, flowing around the global system. Of course this may not be the case in systems where the intelligence is more centralised. Many current commercial autonomic endeavours are at the thicker grained service level.

Granularity is important, e.g. in [21], where unbinding, loading and rebinding a component took a few seconds. These few seconds could be tolerable in a thick-grained component based architecture where the overheads can be hidden in the system's overall operation and potential change is not that frequent. However, in finer-grained architectures, such as operating systems or ubiquitous computing, where change is either more regular or the components smaller, the hot swap time is potentially too high.

One question we may ask is, can systems that provide the same service be compared with each other if the granularity of autonomicity is different? Perhaps at a high level yes.

4.4 Failure Avoidance (Robustness)

Typically many autonomic systems are designed to avoid failure at some level. Many are designed to cope with hardware failure such as a node in a cluster

system or a component that is no longer responding. Some avoid failure by retrieving a missing component. Either way, the predictability of failure is an aspect in comparing such systems. Some systems will be designed for their ability to cope with predicted failure, e.g. using a mean time before failure metric of hardware, and others to cope with unpredictable environments. To measure this, the nature of the failure and how predictable that failure is needs to be varied, and the system's ability to cope with the failure measured. Ability to cope could be in terms of a Quality of Service metric that pertains to the application domain.

For example, in our Kendra³ audio server, which is a closed self-adaptive system, we would test Kendra's failure avoidance abilities by varying the bandwidth in terms of available bandwidth and how quickly that bandwidth varied. This would test its ability to avoid periods of silence given certain environmental circumstances. We observed that Kendra would adapt more gracefully when bandwidth changed little or in a predictable way compared with its operation in a bursty network, which saw Kendra switch frequently between the codecs – sometimes even missing an opportunity to adapt because it did not notice environmental change as it was still handling the previous adaptation [17].

4.5 Degree of Autonomy

Related to failure avoidance, we can compare how autonomous a system is. For example, the NASA pathfinder must cope with unpredicted problems and learn to overcome them without external help. Decreasing the degree of predictability in the environment and seeing how the system copes could measure this. Lower predictability could even mean having to cope with things that it was not designed for. A degree of pro-activity could also compare these features.

4.6 Adaptivity

We separate out the act of adaptation from the monitoring and intelligence that causes the system to adapt. Adaptivity can be something simple as a parameter being changed. Here the adaptation does not impact the performance as much as a component-based reconfiguration. In the latter, a component may need to be hot-swapped, which entails saving its state, locating the new component, binding it into the system and restoring the state from the old component. Some systems are designed to continue execution whilst reconfiguring, while others cannot. Furthermore, the location of such components again impacts the performance of the adaptivity process. That is, a component object which is currently local to the system, versus a component (such as a printer driver for example) which has to be retrieved over the Internet, will have a significantly

³ Kendra is a self-adaptive audio player that was developed in 1996 and adapted the delivery of the audio codec to best suit the available bandwidth between a client and the audio server. It monitored audio delivery and if bandwidth changed another codec was chosen. The aim was to keep the audio quality as best as possible and avoid periods of silence [17].

different performance. Perhaps more future systems will have the equivalent of a pre-fetch of components that are likely to be of use and are preloaded to speed up the re-configuration process.

4.7 Time to Adapt and Reaction Time

Related to cost and sensitivity are measurements concerned with system reconfiguration and adaptation. The time to adapt is a measurement of the time a system takes to adapt to a change in the environment. That is, the time taken between the identification that a change is required until the change has been effected safely and the system moves to a ready state. Reaction time can be seen to partly envelop the adaptation time. This is the time between when an environmental element has changed and the system recognises that change, decides on what reconfiguration is necessary to react to the environmental change and gets ready to adapt. The reaction time affects the sensitivity of the autonomic system to its environment (see next section).

4.8 Sensitivity

This is a measurement of how well the self-adaptive system fits in its environment. At one extreme, a highly sensitive system will notice a subtle change as it happens and adapt (perhaps subtly) to improve itself based on that change. However there is usually some form of delay in the feedback that indicates that some part of the environment has changed. Further, the changeover takes time. Therefore, if a system is highly sensitive to its environment, it can potentially cause the system to be constantly changing configuration and not getting on with the job it has been assigned.

Drawing on our own experience, when measuring Kendra we adjusted its parameters such that the system would become more sensitive. As mentioned in section 4.4, Kendra is a relatively simple self-adaptation system, yet the number of parameters that affected the sensitivity of the adaptation mechanism were many. For example, we could vary the buffer size (which is the data area used to buffer audio), disaster horizon (how close the system thinks it is to a disaster situation), monitoring of sample rates (how much environmental data to monitor and store to predict change in bandwidth). We found that in a generally low bandwidth link, it is better that the system is not sensitive, as that adaptation process impedes too much on the delivery of the sound. However, in good network conditions it is better to be more sensitive, as this delivers the best all round quality of sound [17].

4.9 Stabilisation

Another metric related to sensitivity is stabilisation. This is the time it takes for the system to learn its environment and stabilise its operation. This is particularly interesting for open adaptive systems that learn how to best reconfigure the

system. For closed autonomic systems, the sensitivity is a product of the static rule/constraint set and the stability of the underlying environment the system must adapt to.

4.10 Benchmarking

Finally, it may become necessary to bring these metrics together to form some sort of benchmarking tool. Two approaches can be taken: either we derive new autonomic systems' benchmarks or we augment current benchmarks to incorporate metrics which measure autonomic characteristics.

Our initial attempt involved the Patia project [18]. This project required that we test our autonomic web server and compare its performance with current web servers. We soon found that current web server benchmarks would not be able to test the autonomic aspects of Patia, and in fact they did not measure how web servers were actually being used. It soon became apparent that we would have to design and build a new web server benchmark, we which we called Aeolus [4]. We took research that describes modern web access and data characteristics, and built a benchmark based on this. Further, we wished to test the robustness of our Patia web server under extreme conditions, using flash crowds that would test the autonomic features of Patia to the extreme. Using many of the metrics we have mentioned in this section, we extended the Aeolus web server benchmark accordingly.

Based on our experience, we do not believe that deriving new benchmarks for measuring autonomic systems is the way forward. Instead, due to the diverse applications of autonomic systems, it seems better to augment application-specific benchmarks to include metrics which evaluate autonomic features of that system, e.g. robustness, reaction speed, stability, etc. In particular, the Quality of Service benchmark, which we believe is the top-level measurement of how well the system is meeting its goals, is specific to the application in question. Therefore, we see traditional benchmarks such as the TPC benchmarks being used to measure autonomic DBMSs but perhaps extended to test the autonomous nature of the system.

5 Conclusions

Autonomic computing is an engineering concept that has found its way in a myriad of computing fields. This paper is a review of some typical examples of autonomic computing and attempts to give the reader a feel for the nature of these types of systems, and in doing so illustrate the complexities involved in trying to measure the performance of such systems and to compare them. We have presented two major types of architecture that exhibit autonomic properties and described these as tightly-coupled and decoupled autonomic systems. We have presented the common components found in each of these types of system, and from this derived a set of metrics and methods which we believe are a good starting point to compare autonomic computing systems. These are:

Quality of Service, cost, granularity/flexibility, failure avoidance (robustness), degree of autonomy, adaptivity, time to adapt and reaction time, sensitivity, and stabilisation.

We realise that some of these metrics are more general than others and some pertain to some autonomic systems and not to others. However we believe that the next step is to take this information and derive a more formal method to compare performance of autonomic systems.

A final note regarding our experience of evaluating the Kendra architecture. When testing the system we measured aspects such as general quality levels (audio), as well as unnecessary adaptation, missed opportunities to adapt, sensitivity to environment etc. Kendra is a relatively simple system with closed self-adaptation, yet the performance statistics were of a large volume and difficult to interpret – especially in terms of relating behaviour to varying the many tuning parameters and differing environment conditions. We felt that no concrete quantifiable conclusions were really made other than to say that over sensitivity in bursty networks is bad which we possibly would have guessed. Therefore we imagine the use of data mining techniques to be used to simply understand the volume of performance data presented by such systems.

Nevertheless, it is interesting that to alleviate the maintenance and operation overheads of our modern increasingly complex computing systems, we require the addition of even more complexity. It is our argument that this complexity makes such systems much more difficult to evaluate than before and therefore the need to derive metrics and benchmarks is highly important and interesting.

References

1. J. Appavoo, K. Hui, C. A. N. Soules, R. W. Wisniewski, D. M. D. Silva, O. Krieger, M. A. Auslander, D. J. Edelsohn, B. Gamsa, G. R. Ganger, P. McKenney, M. Ostrowski, B. Rosenburg, M. Stumm, and J. Xenidis. Enabling autonomic behaviour in systems software with hot swapping. *IBM Systems Journal*, 42(1):60–76, 2003.
2. D. F. Bantz, C. Bisdikian, D. Challener, J. P. Karidis, S. Mastrianni, A. Mohindra, D. G. Shea, and M. Vanover. Autonomic personal computing. *IBM Systems Journal*, 42(1):165–176, 2003.
3. J. P. Bigus, D. A. Schlosnagle, J. R. Pilgrim, W. N. M. III, and Y. Diao. Able: A toolkit for building multiagent autonomic systems. *IBM Systems Journal*, 41(3):350–371, 2002.
4. E. N. Bletsas and J. A. McCann. AEOLUS: An extensible webserver benchmarking tool. In *Submitted to the 13th IW3C2 and ACM World Wide Web Conference (WWW04)*, New York City, 17–22 May 2004.
5. E. M. Dashofy, A. van der Hoek, and R. Taylor. An infrastructure for the rapid development of XML-based architecture description languages. In *Proceedings Of the 24th International Conference on Software Engineering (ICSE2002)*, Orlando, Florida, May 2002.
6. E. M. Dashofy, A. van der Hoek, and R. N. Taylor. Towards architecture-based self-healing systems. In *Proc. of the first workshop on self-healing systems*, 2002.
7. D. Garlan and B. Schmerl. Exploiting architectural design knowledge to support self-repairing systems. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, July 2002.

8. D. Garlan and B. Schmerl. Model-based adaptation for self-healing systems. In *Proceedings of the first workshop on Self-healing systems*, Nov. 2002.
9. D. Garlan, B. Schmerl, and J. Chang. Using gauges for architecture-based monitoring and adaptation. In *Working Conference on Complex and Dynamic Systems Architecture, Brisbane, Australia*, Dec. 2001.
10. I. Georgiadis, J. Magee, and J. Kramer. Self-organising software architectures for distributed systems. In *Proceedings of the first workshop on Self-healing systems*, Nov. 2002.
11. J. Kephart and D. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, Jan. 2003.
12. F. Kon, R. H. Campbell, M. D. Mickunas, K. Nahrstedt, and F. J. Ballesteros. 2k: A distributed operating system for dynamic heterogeneous environments. In *9th IEEE International Symposium on High Performance Distributed Computing*, pages 201–210, 2000.
13. S. Kumar and P. R. Cohen. Towards a fault-tolerant multi-agent system architecture. In *Proceedings of the 4th conference on autonomous agents*, 2000.
14. T. Liu and M. Martonosi. Impala: a middleware system for managing autonomic, parallel sensor systems. In *Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming*, June 2003.
15. V. Markl, G. M. Lohman, and V. Raman. LEO: An autonomic query optimizer for DB2. *IBM Systems Journal*, 42(1):98–106, 2003.
16. J. A. McCann. Adaptivity for improving web streaming application performance. *Adaptive Evolutionary Information Systems*, pages 172–191, 2003.
17. J. A. McCann, P. Howlett, and J. S. Crane. Kendra: adaptive internet system. *J. Syst. Softw.*, 55(1):3–17, Nov. 2000.
18. J. A. McCann and G. Jawaheer. Experiences in building the patia autonomic webserver. In *1st International Workshop on Autonomic Computing Systems*, 2003.
19. L. D. Paulson. Computer system, heal thyself. *Computer*, 35(8):20–22, Aug. 2002.
20. D. Pescovitz. Helping computers help themselves. *Spectrum, IEEE*, 39(9), 2002.
21. M. J. Rutherford, K. Anderson, A. Carzaniga, D. Heimburger, and A. L. Wolf. Reconfiguration in the Enterprise Javabeen component model. In *Proceedings of the IFIP/ACM Working Conference on Component Deployment, Berlin, Germany*, pages 67–81, 2002.
22. R. Sterritt and D. Bustard. Towards an autonomic computing environment, 2003.
23. G. Valetto and G. Kaiser. A case study in software adaptation. In *Proceedings of the first workshop on Self-healing systems*, Nov. 2002.
24. G. Valetto and G. Kaiser. Combining mobile agents and process-based coordination to achieve software adaptation. Technical Report CUCS-007-02, Columbia University, Nov 2002.
25. K. Whisnant, Z. T. Kalbarczyk, and R. K. Iyer. A system model for dynamically reconfigurable software. *IBM Systems Journal*, 42(1), 2003.
26. A. Wise, A. G. Cass, B. S. Lerner, E. K. M. Call, L. J. Osterweil, and S. M. S. Jr. Using Little-JIL to coordinate agents in software engineering. In *Automated Software Engineering Conference (ASE 2000)*, Sept. 2000.

Multi-path QoS Routing in TDMA/CDMA Ad Hoc Wireless Networks

Huayi Wu^{1,2}, Chuanhe Huang¹, Xiaohua Jia², and Yanxiang He¹

¹ Computer School, Wuhan University, China

² Dept of Computer Science, City University of Hong Kong
whylei429@hotmail.com

Abstract. This paper investigates the issues of QoS routing in TDMA/CDMA ad hoc networks. Since the available bandwidth is very limited in ad hoc networks, a QoS request will be blocked if there does not exist a path that can meet the QoS requirements, even though there is enough free bandwidth in the whole system. In this paper, we propose a scheme of using multiple paths between two nodes as the route for a QoS call. The aggregate bandwidth of the multiple paths can meet the bandwidth requirement and the delays of these paths are within the required bound. We also propose three strategies for choosing proper paths, namely, SPF, LBF, and LHBF. Extensive simulations have been conducted. The simulation results show that the proposed multiple paths routing scheme significantly reduces the system blocking rates in various network environments, especially when the network load is heavy.

Keywords: QoS routing, QoS call, call blocking, ad hoc networks, TDMA/CDMA.

1 Introduction

A mobile ad-hoc network (MANET) is a collection of wireless mobile nodes that form a temporary network without the aid of any established infrastructure or centralized administration [1]. QoS routing is to find a route that can meet the end-to-end QoS requirements. In this paper, we focus our discussion on two QoS parameters, bandwidth and delay.

In a MANET, each host has very limited bandwidth and energy power, which makes QoS routing much more complicated than that in traditional networks [2]. A call will be blocked when the system cannot find a path that provides required bandwidth, even though the system has enough free bandwidth. We propose a new scheme to find multiple paths whose aggregate bandwidth can satisfy the required bandwidth and whose delays are within the required delay bound. Compared with single path routing methods, our scheme can greatly reduce the system blocking probability and thus make a better use of network resources.

The rest of the paper is organized as follows. Related work is presented in the next section. In section 3, we present the formulation of the problem. Section 4 describes our multiple paths routing protocol. In this section, we propose three strategies to choose the proper multiple parallel paths. We show the simulation results in the section 5. Finally, section 6 concludes this paper.

2 Related Work

Many routing algorithms have been proposed for ad hoc networks. 1) Pro-active algorithms where a routing table is maintained at every node, such as the DSDV [3] and the ADV [4]; 2) On-demand algorithms in which a route is discovered on-demand whenever there is a call setup request, such as the AODV [5] and the DSR [6]; 3) Virtual backbone methods where a virtual backbone of a network is maintained for connecting a source to a destination, such as the CEDAR [7] and Spine-based method [8].

There are also a lot of works on multi-path routing in the literature such as [2], [9], [10], [11], [12], [13], [14]. However, almost all the previous works are for the purpose of fault tolerance or failure recovery. That is, there is always a primary path, plus a set of backup paths. When the primary path fails during a communication session, one of the backup paths will take the role as the new primary path. The studies in [13] mainly use multiple paths to deliver multiple secret message shares in order to enhance security. Different from them, our scheme uses the multiple paths in parallel, so that the aggregate bandwidth of them can meet the bandwidth requirement of a single call and the delays of these paths are all within the required delay bound.

3 Problem Formulation

In this paper, we assume the MAC sub-layer adopts the CDMA-over-TDMA channel model [14, 15 and 16]. In CDMA, each node uses a pre-assigned code for communication with neighbours in a conflict free fashion [18]. Hence we do not need to consider transmission conflicts or interference among the nodes.

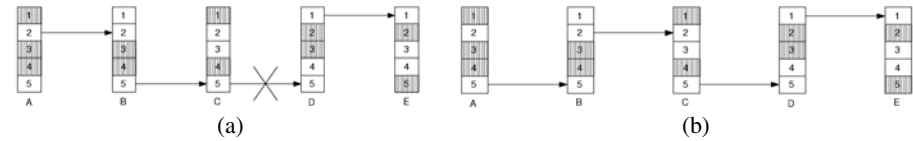


Fig. 1. Two examples of timeslot assignment (timeslots in grey are occupied)

In this paper, bandwidth is measured in the unit of the free timeslots. Notice that, the free bandwidth over a path depends not only on the free timeslots over the links in the path, but also on the slot assignment method. For example in Fig. 1(a), the slot assignment on link $\langle B, C \rangle$ conflicts with the assignment on link $\langle C, D \rangle$, because slot 5 was assigned to both links and node C cannot do both receiving from B and transmitting to D simultaneously at slot 5. Fig. 1(b) shows a good case of slot assignment, which provides one unit of bandwidth for the path. In this paper, we assume the slot assignment algorithm in [15] is used.

The goal of our routing scheme is to reduce system blocking rate by using multiple paths in parallel. At the same time, the aggregate bandwidth of the multiple paths can meet the bandwidth requirement and the delays of these paths are within the required bound. Our scheme has three steps. The first step is the route discovery process, the second is the timeslot assignment and reservation and the third is the selection of

multiple parallel paths. We propose three strategies for selecting proper multiple paths.

4 Multiple Paths Routing Protocol

4.1 Route Discovery

As our routing scheme is based on the “on-demand” protocol, the source node floods route-request packets to discover the routes to the destination only when it is necessary. Unlike some protocols that take bandwidth-reservation into consideration at the stage of the route discovery, we firstly build a connection and find candidate paths in parallel; secondly, choose proper multiple paths at the destination node by calculating the bandwidth and the delay of each path and lastly reserve the necessary bandwidth over each chosen path for the preparation of the data transmission.

The operations of the route discovery are as follows. When a source node receives a request from the application layer to set up a QoS call to a destination node, with the bandwidth requirement B and maximal delay bound D , it prepares a route-request (RR) packet by setting the TTL value to D and floods the RR packet to its neighbors. The packet contains the following fields: (source, destination, seq-ID, type, route, freeslots, B , TTL), where (source, seq-ID) is used to uniquely identify a packet. The “seq-ID” is a sequence number which can be used to check duplicate copies of an old request and detect the stale cached routes. The “type” refers to the packet type that may be RR or RP (route reply). The field “route” records the routing information and the field “freeslots” records the information of free slots at each node in the route.

When a node receives a RR packet, if the pair (source, seq-ID) of it was seen before, it discards this packet and does not pass it further. Otherwise, it checks if there is any common free timeslot between this node and the last hop-sender. If not, it means there is no bandwidth to receive from the last node in the path and the RR is dropped. Otherwise, this node will further flood the RR packet out if it is not the destination. It first decreases “TTL” field by one. If TTL counts down to zero, it means this RR packet has gone outside of the intended routing range and the packet is dropped. It then adds to the RR packet its own address to the “route” and its free timeslot information to the “freeslots”. It finally floods this RR packet out. This operation is repeated node by node until the RR packet reaches the destination.

4.2 Policies for Selecting Multiple Parallel Paths

After the route discovery, the destination has the information of all the paths to it and it can calculate the bandwidths of them. The next step is to choose the suitable paths whose aggregate bandwidth can meet the bandwidth requirement of the request. As shown in Fig. 3, each of path 1 and path 2 has one unit of bandwidth (dotted lines connect the unused free timeslots). If there is a request to setup a call between A and F that requires two units of bandwidth, we need to use the both path in parallel to meet the bandwidth requirement of the call.

We propose three strategies for selecting the proper multiple paths, which will be discussed in the following subsections.

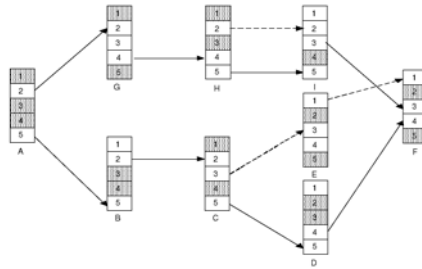


Fig. 2. Two parallel paths from A to F (slots in grey are occupied)

4.2.1 Shortest Path First (SPF)

Shortest path routing is the default routing method used in most of the networks. Shortest path routing uses the shortest path between the source and the destination as the route, which incurs short delay and costs less network resources (the data travels less distance in the network). In the SPF method, we sort all the candidate paths according to their lengths (in terms of hop-counts) in ascending order. Then, we take the first group of paths whose aggregate bandwidth can just meet the required bandwidth of the call. By using this method, the average delay of the selected paths should be the minimal, but it may use too many paths to meet the required bandwidth because some shortest paths may provide only a small amount of bandwidth.

4.2.2 Largest Bandwidth First (LBF)

Contrast to the SPF method, the largest bandwidth first method aims at reducing the number of parallel paths required. In the LBF method, we sort the candidate paths according to the available bandwidth they provide, and choose the paths that have the largest available bandwidths and whose aggregate bandwidths meet the requirement of the call. By using this method, the number of paths selected should be minimal, but a selected path might have a long distance because the path needs to take lightly loaded links in order to provide more bandwidth. This may cause two problems: a) the longest path will pull down the delay of the call; and b) the data packets may eventually cost more network resources due to the long distance they travel.

4.2.3 Largest Hop-Bandwidth First (LHBF)

As we have seen the advantages and disadvantages of the SPF and LBF methods, the largest hop-bandwidth first method aims at striking a balance between the two methods. We define the hop-bandwidth of a path as the value of available bandwidth of the path divided by its hop-counts. Thus, hop-bandwidth represents the amount of bandwidth per hop, provided by a path. In this method, we sort all the candidate paths in descending order according to their hop-bandwidths, and then select the first group of paths whose aggregate bandwidth can just meet the requirement of the call. By using this method, the average hop-bandwidth is the maximal, which means the call will cost less network resources for data transmission.

5 Simulations

Since each of the proposed three path selection strategies has its own aims and shortcomings, the simulations are designed to evaluate the performance of the three strategies under various network situations. The performance is evaluated in three aspects: a) blocking rate of the system, b) the number of paths selected for a request and c) average cost of network resources.

5.1 Simulation Setup

The simulation is conducted in a 100×100 2-D free-space by randomly allocating N nodes ($N = 100$). The radius of transmission range of all nodes is set to be the same 30 throughout the simulation process. Once the nodes are placed in the square region and their transmission range are decided, a network graph is formed where two nodes within each other's transmission range will have a link. Any unconnected graph will be discarded. The number of timeslots at each node is set to be 16. The network load is defined as the average percentage of occupied timeslots in all nodes in the system, which varies between 0 and 1. During the simulations, we randomly generate traffics and inject them into the network to make the network load at a specified level.

Throughout the simulations, a QoS call setup request is generated as follows. The source and destination nodes are randomly picked up from the network graph. We assume that they are not neighbors. The delay bound of the call is set to be twice as the hop-counts of the shortest path between the source and the destination. We simulate three types of requests, namely low-bandwidth, medium-bandwidth and high-bandwidth requests, whose bandwidth requirements are set to 2, 5, and 8 timeslots, respectively. The values in the following figures are the average ones of 100 runs. Each time, a request is generated as above and all the routing algorithms are executed.

5.2 Simulation Results and Analysis

In the first experiment, we compare the blocking rates of the three proposed algorithms. We introduce a single path (SP) routing protocol as a performance benchmark. In the SP protocol, only the shortest path that has the required bandwidth has been chosen as the route if there is such a path.

The simulation results are shown in the figures (Fig. 3(a) -Fig. 3(c)). From the figures, the following observations can be made:

- 1) The multi-path routing scheme greatly reduces blockings of the requests in all network load situations. This reduction becomes even more significant when the network load is heavy or the bandwidth requirements are high. By using SP method, the blocking rate reaches the ceiling (100%) quickly with the increment of network load.

- 2) LBHF method performs better than the other two multi-path methods. The main reason is that the LBHF method minimizes the actual network resources for each request. In a long run, the system will have more resources for later requests, which makes the LBHF method overall has less blockings than the other methods. Notice that the curves of SPF and LBF overlap with each other in all three figures in Fig. 3.

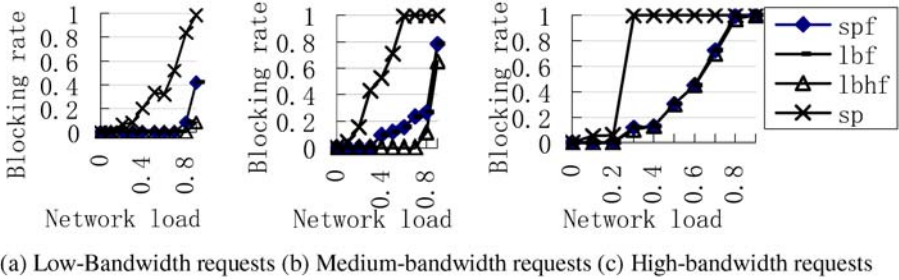


Fig. 3. Blocking rate versus network load

In the second experiment, we study the numbers of selected multiple paths in the three proposed methods.

The simulation results are shown in Fig. 4. From the figures, the following observations can be made:

1) With the increase of network load, the numbers of selected multiple paths in the three proposed methods become greater. This is because when the network load is light, it is easier to find a single path which satisfies the bandwidth requirement. The heavier the network load is, the less the available resource on each single path and the more paths are needed to make the aggregated bandwidth meeting the requirement.

2) The numbers of multiple paths in the methods of LBF and LBHF are close and smaller than that in the SPF method. As we discuss in the first experiment, the system can save more resources in a long run in the LBHF method and less paths are needed. Besides, in LBF and LBHF methods the bandwidth is the main factor in selecting multiple paths. So they perform better than the SPF method in the number of paths in use. This performance gap is narrowing when the network load becomes heavy, because almost all the available network resources are utilized in this case.

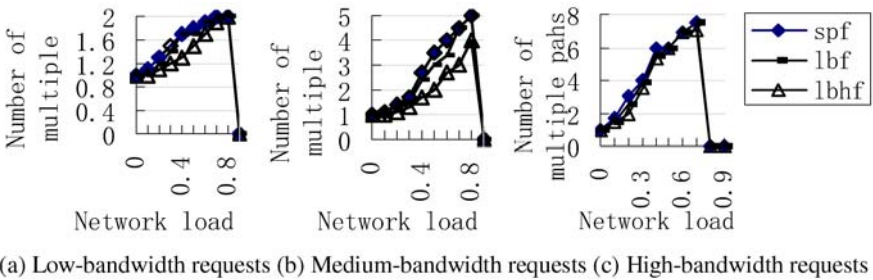


Fig. 4. Number of parallel paths versus network load

In the third experiment, we first define the cost of network resources of a QoS call. $P = \{p_1, p_2, \dots, p_k\}$ is the set of selected paths that are used by the call for data transmission. The hop-count (p_i) and bandwidth (p_i) denote the number of hop-count and the value of bandwidth of Path p_i , respectively. The cost of network resources for a call is defined as (1).

From the definition (1), we can see that the network cost of a call represents the actual network resources consumed by the call. When the network resources are sufficient, the size of the bandwidth of a path may exceed the requirement. We only consider the actually used bandwidth by transmitting signals.

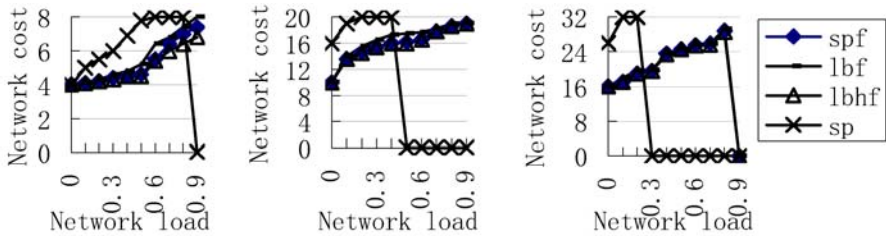
$$\text{NetworkCost} = \sum_{p_i \in P} \text{hop-count}(p_i) \times \text{bandwidth}(p_i) \quad (1)$$

The simulation results are shown in Fig 5. From the figures, the following observations can be made:

1) The cost of network resources increase more gently in the multiple paths routing scheme than in the single path scheme as the network load becomes heavier. Since the results in the figures present the network cost of successful QoS calls, when all the QoS calls are blocked, the cost of the calls reaches zero. These figures again show that the system blocking rate is much lower in our multiple paths scheme than in a single path scheme.

2) At the same level of network load, the cost of network resources is less in the multiple paths routing scheme than in the single path routing scheme, because the goals of our three strategies are making the hop-count small or making the sum of products of bandwidth and hop-count of every path small.

3) The LBHF method performs better than the other two multi-path methods in network cost of a call. This matches the original goal of the LBHF method, which minimizes the cost of network resources for each request.



(a) Low-bandwidth requests (b) Medium-bandwidth requests (c) High-bandwidth requests

Fig. 5. Network cost versus network load

6 Conclusions and Discussions

We have discussed the QoS routing in TDMA/CDMA ad hoc wireless networks. A new scheme that uses multiple paths in parallel to meet the QoS requirements of a call has been proposed. It has two major advantages: 1) it greatly reduces the system blockings. Thus, system resources can be better utilized; 2) the proposed routing protocol follows the format of existing on-demand routing protocols for ad hoc networks, which makes it easy to be incorporated into the existing single-path routing systems.

We also proposed three strategies for selecting multiple paths as the route of a call, namely, SPF, LBF and LBHF. Each of them has a different objective, such as minimizing the delay of a call, minimizing the number of paths in use, or minimizing the

overall network cost. The three strategies can be used in different network environments and for meeting different application needs. Extensive simulations have been conducted to evaluate the performance of the proposed scheme. Simulation results have demonstrated the effectiveness of our method in reducing the network blockings.

References

1. T. Camp, J. Boleng, B. Williams, L. Wilcox and W. Navidi, "Performance comparison of two location based routing protocols for ad hoc networks", *Proceedings of IEEE INFOCOM*, pp.1678-1687, 2002.
2. Yuh-Shyan Chen, Yun-Wen Ko, and Ting-Lung Lin, "A Lantern-Tree-Based QoS Multicast Protocol with Reliable Mechanism for Wireless Ad-Hoc Networks", *Proceedings of IEEE ICCCN'02*, Oct. 14-16, 2002, Miami, Florida, U.S.A., 2002.
3. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer Communications Review*, pages 234-244, October 1994.
4. R.V.Boppana and S.P.Konduru, "An Adaptive Distance Vector Routing Algorithm for Mobile, Ad Hoc Networks", *Infocom*, pp.1753 - 1762, vol.3, April 2001.
5. C. E. Perkins, E. M. Royer, and S. R. Das. "Ad hoc On-Demand Distance Vector (AODV) Routing". <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt>, Jan 2002. *IETF Internet Draft* (work in progress).
6. D. B. Johnson, D. A. Maltz and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.
7. R. Sivakumar, P. Sinha and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm", *IEEE Journal on Selected Areas in Communication*, Vol. 17 (8), 1999, pp. 1454-1465.
8. R. Sivakumar, B. Das, and V. Bharghavan. "An improved spine-based infrastructure for routing in ad hoc networks", *Proceedings of the International Symposium on Computers and Communications (ISCC'98)*, 1998.
9. I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing", *IEEE/ACM Transactions on Networking*, vol. 7, pp. 885-896, Dec. 1999.
10. A. Nasipuri and S.R. Das., "On-demand multi-path routing for mobile ad hoc networks", in *IEEE ICCCN'99*, pages 64-70, IEEE, 1999.
11. S.J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc network", in *IEEE WCNC'00*, pages 1311-1316, IEEE, 2000.
12. P. Pham and S. Perreau, "Performance Analysis of Reactive Shortest Path and Multi-path Routing Mechanism with Load Balancing", *IEEE INFOCOM'03*.
13. W. Lou, W. Liu and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks", *IEEE INFOCOM'04*.
14. Y. Ganjali and A. Keshavarzian, "Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multi-path Routing", *IEEE INFOCOM'04*.
15. C. R. Lin, "Admission control in time-slotted multihop mobile networks", *IEEE Journal on Selected Areas in Communications*, Volume: 19 Issue: 10, Oct. 2001; Page(s): 1974-1983.
16. C. R. Lin, "On-Demand QoS Routing in Multihop Mobile Networks", *Proceedings of IEEE INFOCOM 2001*, pp.1735-1744, 2002.
17. C. R. Lin and J. S. Liu, "QoS Routing in ad hoc wireless networks", *IEEE J. Select. Areas Commum*, vol.17, pp1426-1438, Aug.1999.
18. Limin Hu, "Distributed Code Assignment for CDMA Packet Radio Networks", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 6, pp. 668-677, Dec. 1993.

Economic Heuristic Guided Price-Regulating Mechanism in SHGRB*

Tang Jun, Tong Weiqin, Zhi Xiaoli, and Yin Zhijie

School of Computer Engineering and Science,
Shanghai University, Shanghai, 200072, P.R. China
tangfly_jj@163.com, {wqtong,xlzhi}@mail.shu.edu.cn

Abstract. The accelerated development in Grid Computing has positioned as promising next generation computing platforms for solving large-scale resource intensive problems. However, the management of resources and scheduling computations in a Grid environment remain complex and immature. The computing economy is argued for the necessity to create a real world scalable Grid because it provides a fair basis in successfully regulating decentralization and heterogeneity presented in Grid environment. Although efforts have been made on the economic mechanisms in Grid, the fixed cost model of resources pricing in current super-scheduler or meta-scheduler and the weakness of load balance in resource scheduling should be improved by any means. Therefore, this paper proposes an economic heuristic guided price-regulating mechanism in the Shanghai Grid Resource Broker (SHGRB) to 1) better adapt to the dynamic changes of grid environment; 2) regulate resource prices for stronger load balance; 3) provide higher quality services for users.

1 Introduction

Computational grids [1] are becoming attractive and promising platforms for solving large-scale applications of multi-institutional interest. However, the management of resources in a Grid environment becomes complex. The geographic distribution of resources that are heterogeneous in nature, owned by different organizations with their own accesses policies and cost models, and have dynamically varying loads and availability introduce a number of challenging issues that Grid resource management systems need to address.

Fortunately, using economy idea for resource management and scheduling throws light on treating with this issue, which provides a fair mechanism in successfully managing decentralization and heterogeneity that is also presented in human economies. Yet computing economy is rarely taken into consideration in the design of the most current systems such as Globus [2], NetSolve [3], AppLes [7], etc. Although the idea of economy was used in the scheduling in Nimrod/G [4][5][6], there are some shortages which are: (1) the fixed cost model to determine where a submitted task to be executed; (2) the heavy load of cheap resources using cost optimization algorithm.

* This work is support in part by the Science and Technology Committee of Shanghai Municipality under grant No. 03dz15026.

Such model is unable to adapt to the changing load and availability of resources in the Grid environment for realizing economy idea.

The price charged to customers by a resource provider may be varied from time to time even from users to users. A flexible pricing model for a resource is of vital importance. On one hand, resource price should be able to fluctuate guided by the economic heuristic according to the current load status. On the other hand the regulated price also should be able to influence its load changing.

Therefore, this paper proposes an economic heuristic guided price-regulating mechanism in the Shanghai Grid Resource Broker (SHGRB) based on the work in the Shanghai Grid testbed. The architecture of SHGRB is divided into two independent parts (scheduler advisor and pricing agent) centrally controlled by Job Control Agent (JCA).

The organization of this paper is as follows. In the next section the architecture of SHGRB is introduced and the model of price regulation is proposed in section 3. Section 4 presents the economic heuristic guided pricing agent in details. Section 5 discusses and analyzes the algorithm and experimental testing. Finally comes the conclusion.

2 The Architecture of SHGRB

Resource Broker (RB) is a global resource management and scheduling system that supports economy-based computations in Grid computing environments for parameter sweep application. This grid scheduler, also known as super-scheduler or meta-scheduler, is different from that of the local domain (a single cluster or supercomputer) for its performing task schedule at a higher layer of system-level middleware toolkit/scheduler (like Globus) services.

RB uses scheduling algorithms or policies for mapping tasks to resources to optimize system or users objectives according to their goals. This is an essential part in resource management architecture, since it influences the effectiveness of the resource management strategy directly. *RB* also masks the complexities of grid environment to users, when it acts as a mediator to discover, negotiate, and select resources, map tasks to selected resources, start and monitor the execution, record the middle results if necessary, finally collect and return the ultimate results to users. What users need to do is only to submit their tasks to the resource broker with their requirements presentation and then waiting for the results. The interaction between users and resource broker is shown in Fig 1.

Fig 2 illustrates the architecture of SHGRB.

The key components of SHGRB are: Job Control Agent (*JCA*), Grid Explorer (*GE*), Resource Advisor (*RA*), and Pricing Agent (*PA*).

JCA: this component is a persistent central component responsible for controlling and switching between scheduler advisor and pricing agent and shepherding a job through the system and the creation of jobs, monitoring job status, interacting with users, schedule advisor, and dispatcher.

GE: this component is responsible for resource discovery by interacting with grid-information server and identifying the list of authorized machines, and keeping track of resource status information.

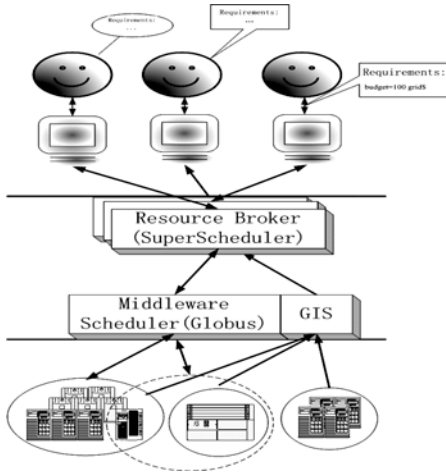


Fig. 1. The interaction between users and RB

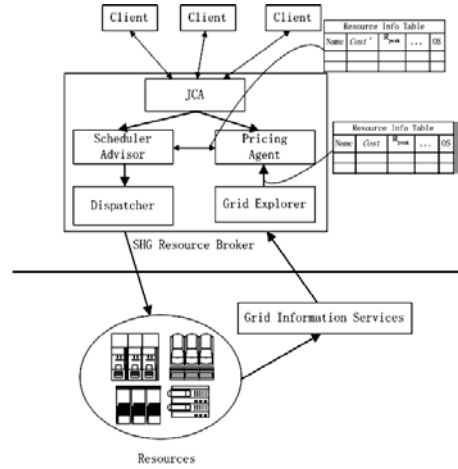


Fig. 2. The architecture of SHGRB

RA: this component is responsible for resource discovery using *GE*, resource selection, and job assignment (schedule generation). Its key function is to select resources that meet user requirements while assigning jobs to resources.

PA: this component is responsible for resource price regulating according to the economy heuristic and resources current load acquired from *GE* component at every regulating phase of process step, and maintaining or updating the Resource Information Table (*RIT*) for the scheduler advisor to make resource selection.

As Fig 2 shows, our motivation of designing *RIT* is to (1) aggregate resources information distributed in GIS; (2) store the regulated prices; (3) speed up resource selection by the scheduler advisor. Advisor acquires information directly from its local *RIT*, when it starts to regulate prices or search for resources availability.

3 The Model of Price Regulation

For regulating resource prices at proper time, the concepts of “*scheduling phase*” and “*regulating phase*” are introduced. As Fig 3 shows, the structure of *RB* arises from the progress of its work through time. For the whole work of *RB*, this is a sequential composition of global steps. Each process consists of two phases and four notification events as following:

- (1) *JCA*→*SA* notification event: *JCA* triggers *SA* to start up jobs scheduling.
- (2) *Scheduling Phase*: This phase includes the process of initiating, scheduling, executing, or result collecting etc., during which the price of resources is stable. The scheduler advisor makes resource selection decision using the price in its local *RIT*.
- (3) *SA*→*JCA* notification event: *SA* informs *JCA* of its work finishing.
- (4) *JCA*→*PA* notification event: *JCA* triggers *PA* to start up price regulation.
- (5) *Regulating Phase*: the advisor triggers the *PA* to work. After retrieve the current state and information of resources from GIS, the agent recalculates the price ac-

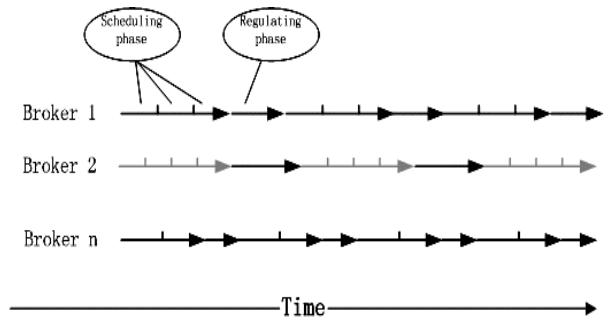


Fig. 3. The work of RB

cording to users or default economic regulation heuristic; finally, updates *RIT* for the local scheduler advisor.

(6) *PA*→*JCA* notification event: *PA* informs *JCA* of its work finishing.

These steps iterate during the work of SHGRB.

A vital factor influencing the performance of scheduling and pricing agent is the interval of every two regulating time. On one hand, if it is too long, the outdated or invalid information in *RIT* will lead to frequent failures for resource selection; on the other hand, if it is too short, the performance will be subjected to decline due to the increase overhead caused by the frequent regulating.

4 The Economic Heuristic Guided Pricing Agent

The price fluctuation should be able to reflect the dynamic changes of the current supply and demand status, so that allocation optimization and equilibrium could be reached. The law of demand explains the inverse relation between demand and price in general. It can be stated as follows:

"*Ceteris Paribus* (other things remaining equal), the quantity of a goods demanded will rise (expand) with every fall in its price and the quantity of a goods demanded will fall (contract) with every rise in its price." In order to satisfy price-demand relation, the effect of other variables has been restrained by assuming them to be constants [8].

The relationship between demand and price can be showed in the form of demand schedule and demand curve, Fig 4 and Fig 5 respectively.

Take advantage of the price-demand relation to regulate resource price. Raise resource price to decrease its demand, when its load reaches a certain high threshold; and drop resource price to increase its demand, when its load reaches a certain low threshold.

Fig 6 (a) and (b) are illustrated two general regulating heuristics respectively. This paper adopts the heuristic shown in (b), whose functional form can be stated as,

D(demand) q_d	Schedule Price
10	0
8	1
4	2
1	3
0	4

Fig. 4. Demand schedule

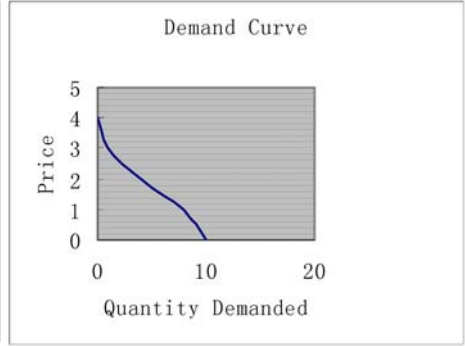


Fig. 5. Demand Curve

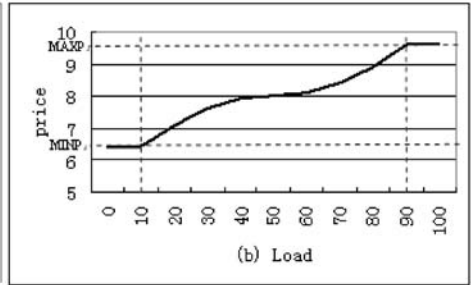
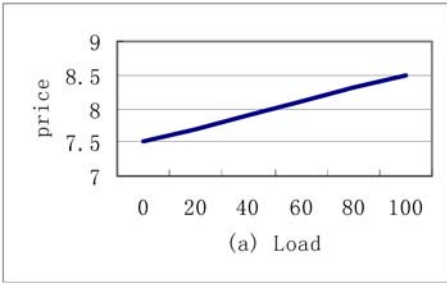


Fig. 6. Regulating heuristic

$$P = \begin{cases} a + ((x-b)*10)^2/c & x \geq b. \\ a - ((x-b)*10)^2/c & 0 < x < b. \end{cases}$$

$$MINP \leq P \leq MAXP.$$

Variable x represents the current load of a resource. Constant a , b , c is defined by the providers or agents. Usually, constant a represents the basis or initial price of a resource; b represents the valve value of resource load for price increase or decrease; c controls the intensity degree of price changing.

Resource owners should be able to define $MINP$ and $MAXP$ as the lowest and highest price threshold respectively, so that resource owners profits can be guaranteed, which can be noted from Fig 6(b).

5 The Algorithm and Experiment Testing

We adopt the rescheduling strategy to better adapt to the changes of grid environment. At each scheduling phase super-scheduler acquires the resource status information

from *RIT* and schedules the jobs that are new or mapped but not started to be executed, so it is possible one job may be scheduled for more times. Although such re-scheduling strategy may increase system overhead, it is able to have more opportunities for better job-resource pair and gaining more benefits due to the ever updating of the resource information and the complex, dynamic computational grid environment.

With the centrally controlled support of *JCA*, the algorithm embedded in *SHGBR* is divided into two parts: (1) resource selection heuristic for scheduler advisor; (2) price regulating heuristic for regulating agent.

Following is the algorithm for scheduler advisor with cost-optimization algorithm:

1. RESOURCE ACQUIRING: identify the available resources and their characteristics from *RIT* and sort them by the increasing order of their prices
2. RESOURCE SCHEDULING: repeat while there exists unprocessed jobs and current time is within the deadline
 - for each task to schedule
 - assign the job to the cheapest resource
 - remove the job from the unprocessed jobs
3. JOB DISPATCHING: identify the number of jobs without overloading the resource

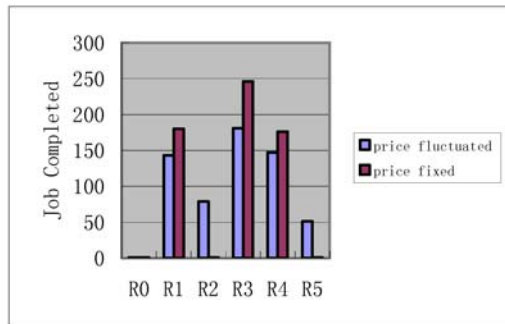
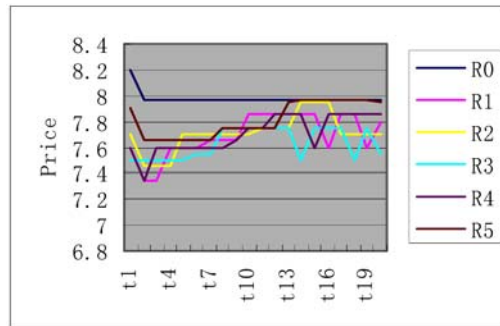
The algorithm for price regulating heuristic of Fig 6(b) is as follows:

1. RESOURCE ACQUIRING: identify the available resources and their characteristics from *GIS*
2. RESOURCE REGULATING: for each resource
 - identify its load and price
 - calculate P with heuristic of Fig 6(b)
3. TABLE UPDATING:
 - update *RIT* with the regulated price
 - delete the departure resources from *RIT*
 - insert the new joined resources into *RIT*

To identify the performance of *SHGRB*, a simulation experiment is made based on the resources in the Shanghai Grid testbed environment. For simplicity, all sub-tasks/jobs are assumed to be independent and have no communications and data exchanges with each other. In this experimental testing, a modeled of task farming application is established which consists of 600 jobs packaged containing the parameters of job length, the size of job input and output data along with other parameters needed by the *RB*. We simulated 6 resources with various characteristics, capabilities and configurations as those in the *SHG* testbed shown in Table 1, from which it can be noted R3 is the cheapest one followed by R1 and R4, and R0 is the most expensive resources followed by R5. The broker needs the parameter of cost per jobs in terms of G\$ for each resource, which is useful for the identification of the cost of resources. We assume that the cost of resources shown in Table 1 is the price of middle-load. So for the heuristic shown in Fig 6 (b), constant b may be 0.5, and constant a may be the costs listed in the 7th item of Table 1, while c is assumed as 100 for prices fluctuating in a small scale.

Table 1. Grid resources simulated in GridSim

Resource Name	Resource ID	Operating System	Storage capability (TB)	R_{peak} (GFlops)	Bandwidth (M)	Base Cost Per job (G\$)
SHU	R0	Redhat 8.0	2	450	2000	8.2
SSC	R1	Redhat 7.3	1.28	384	100	7.6
SJTU(1)	R2	Redhat 9.0	4	64	1250	7.7
SJTU (2)	R3	Redhat 9.0	1	72	1250	7.5
TJU	R4	Redhat 8.0	0.304	106.24	100	7.6
SUTIC	R5	Redhat 8.0	unknown	unknown	100	7.9

**Fig. 7.** Jobs completed on each resource**Fig. 8.** Price fluctuating during schedule time of each resource

The number of the jobs completed on each resource is illustrated in Fig 7.

It can be observed from this figure, as R1, R3 and R4 are the cheapest resources, they are taking on all resources execution with full load in price fixed model of non-regulating *RB*, while in price fluctuated model the broker of price regulating also allocates jobs to the non-loading resources such as R2, and R5 to share the burden when reaching the acceptable prices (with the prices increase of R1, R3 and R4 and prices decrease of R2, and R5). For the entire two price models, R0 hasn't been allocated any jobs, due to its expensive basic cost leading to less competition even after price decrease regulating. This means the costs provided by the resource owners have a great influence on the resource selection. This can be visible directly from Fig 8, as

R0 is still the most expensive resources even after price is regulated by the price-regulating agent. Meanwhile Fig 8 also illustrates the price fluctuation process of all the resources. It also can be noted that the prices of R1, R3 and R4 have fluctuated around their basic costs adapting to the changing of their loads.

6 Conclusion

In this paper an economic heuristic guided price-regulating mechanism in SHGBR is presented. The goal is to regulate resource prices for a certain load balance to better adapt to the dynamic changes of grid environment and improve the quality of services of resource selection for users. By employ price-demand relation in economy theory, the price can be regulated according to the current resource load, to relief or increase resource load. Experimental testing is also made based on the Shanghai Grid testbed.

References

1. Foster and Kesselman, C. (ed.): The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA (1999)
2. Globus Project – <http://www.globus.org>
3. Casanova, H., Dongarra, J.: NetSolve: A Network Server for Solving Computational Science Problems, Intl. Journal of Supercomputing Applications and High Performance Computing, Vol. 11, No. 3 (1997)
4. Buyya, R., Chapin, S., DiNucci D.: Architecture Models for Resource Management in Global Computational Grids, <http://www.buyya.com/ecogrid/>
5. Buyya, R., Abramson, D., Giddy, J.: Economy Driven Resource Management Architecture for Computational Power Grids, International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2000), Las Vegas, USA (2000)
6. Buyya, R., Abramson, D., Giddy, J.: Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid, 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000), Beijing, China (2000)
7. AppLes Project – <http://apples.ucsd.edu/>
8. Shuang, L., Genhong, Z.: Accounting English, China Financial & Economic Publishing (1996)

Workflow-Based Grid Portal for Quantum Mechanics*

Sung-Wook Byun, Yun-Kyoung Lee, Yong-Won Kwon,
So-Hyun Ryu, and Chang-Sung Jeong

Department of Electronics Engineering, Graduate School, Korea University
5-ka, Anam-Dong, Sungbuk-ku, Seoul, Korea
{sungwook,luco,messias}@snoopy.korea.ac.kr, apple7224@korea.ac.kr,
csjeong@charlie.korea.ac.kr

Abstract. We have been developing Workflow-based grId portal for problem Solving Environment for a Quantum Mechanics (QMWISE). Quantum mechanical calculation is a common and essential method in computational chemistry, however, very intensive and time consuming. Thus, we propose a convenient and effective way to ease the current quantum mechanical problems with Grid technologies. Also, we propose a new workflow-based quantum mechanical portal to approach the difficult quantum mechanical calculations easily by the functions to watch and control the calculation process in the run time, and to manage the large data.

1 Introduction

In these days, computational chemistry is expected to play a major role in fields such as computer-aided chemistry, pharmacy, and biochemistry. Computational chemistry is used in a number of different ways. One particular important way is to model a molecular system prior to synthesizing that molecule in the laboratory. Although computational models may not be perfect, they are often good enough to rule out 90% of possible compounds as being unsuitable for their intended use. This is very important because synthesizing a single compound could require months of labor. A second use of computational chemistry is in understanding a problem more completely. There are some properties of a molecule that can be obtained computationally more easily than by experimental means. There are also insights into molecular bonding, which can be obtained from the results of computations that cannot be obtained from any experimental method.

Computational chemistry now encompasses a wide variety of areas, which include quantum chemistry, molecular mechanics, molecular dynamics, Monte Carlo methods, Brownian dynamics, continuum electrostatics, reaction dynamics, numerical analysis methods, artificial intelligence, chemometrics and others. In this paper, we focus on quantum chemistry.

* This work has been supported by a Korea University Grant, KIPA-Information Technology Research Center, University research program by Ministry of Information & Communication, and Brain Korea 21 projects in 2004.

Quantum chemistry is the determination of various properties of molecules using the principles of quantum mechanics, the central equation is Schrödinger Equation,

$$H\Psi = E\Psi \quad (1)$$

Where H is the Hamiltonian, which incorporates the nuclear kinetics and potential energy terms as well as the electronic kinetics and potential energy terms, Ψ is the wave function, which is a function of nuclear and electronic coordinates and contains all of the information about the system, and E is the total energy. Molecular properties that can be calculated by solving Eq.(1) are the molecular geometry, relative stabilities, vibrational spectra, dipole moments, reactivity, and atomic charges to make a few. However Eq.(1) cannot be solved exactly for atomic and molecular systems, so various approximations are employed. These approximations are divided into two fundamentally distinct groups. One groups are concerned with purely non-empirical methods, so called ab-initio methods. Ab-initio calculations on small to medium sized molecules require a significantly amount of computer resources. Three commonly used ab-initio programs are GAMESS [1], Gaussian [2], and NWChem [3]. The second groups are semi-empirical methods. For larger molecules ab-initio methods require extensive computer resources. It is well known that the difficulty in performing ab-initio calculations on large molecules with a modest basis set is that the number of two-electron integrals that are needed is overwhelming. To overcome some of the computational difficulties, semi-empirical methods are made in which several of the integrals are parameterized or neglected. The most widely used software package that incorporates these approximations is known as MOPAC [4]. These programs implement the MINDO, INDO, MNDO, AM1, and PM3.

These software programs are very powerful tools, however, typically limited by the reasons as follows: (1) Not easy to use these tools because of the need of specific knowledge of the usage of the tools (2) Impossible to watch and control the calculation process while the process is running (3) Cannot predict the time for calculation (4) Not provided graphics user interface (5) Limited by calculation capabilities (6) Limited by accessibility of a remote user (7) Difficult to manage parallel computation. To solve these problems, computational chemistry in the era of Grid technology is very challenging. Thus, we will propose workflow-based computational Grid portal for quantum mechanics (QMWISE) in this paper.

The remaining sections of this paper describe the previous work of quantum chemistry on Grid environment in section 2, a simple outline of WISE system in section 3, QMWISE in section4, finally, conclusions and future work.

2 Previous Work

There were some efforts to improve environments of quantum mechanics calculations. GAMESS web portal services were proposed by Kim K. Baldridge et al [5]. They purposed development of computational chemistry web portal, XML schema based on output data of electronic structure software, and database and

associated query tools that will serve as a basis for storage, retrieval, and manipulation of QM data in uniquely new ways. Initially, GAMESS portal was to isolate users from the complexities of the computational grid by providing a standard user interface for accessing input and output, running jobs on one of a variety of platforms without logging onto those platforms, and the ability to transfer data among various platforms. Also, the portal has facilities for processing the output of a particular run via visualization with their computational chemistry visualization and analysis tool, QMview.

Also, design and implementation of the intelligent scheduler for Gaussian Portal on quantum chemistry Grid (QC Grid) are investigated by T. Nishikawa et al [6]. QC Grid consists of a Web interface, a meta-scheduler, computing resources, and archival resources on Grid infrastructure. The QC Grid user has a web-based interface that does all operations such as uploading the input file, controlling the job, displaying job status, and getting results. The web-based interface was created by using a toolkit that can quickly build the portal interfaces. It is served by an HTTP server daemon program. The HTTP server supports Secure Socket Layer (SSL). The web interface is used to request job resources, control job status, visualize molecular structures, and display results. The meta-schedulers modules include job scheduling functions and enables optimal allocation of the entire computing resource for a large number of jobs. The Grid infrastructure ensures security, management of resource allocation, access to remote data, and monitoring of remote resources.

Both classes of grid computing of quantum chemistry are not considered the process of designing workflow based web services. Workflow is used to express a complex process as a set of interconnected, smaller, less complicated component tasks [7]. In various areas such as industrial and administrative process management and design processes, workflow has successfully been applied. Also, there are many commercial workflow management systems and research prototypes developed for various applications [8]. These classes of workflow systems have not tried to customize for quantum chemical calculations. In addition, these workflow systems are not designed to work with computational Grids. In Grid computing area, there are currently efforts on application-specific groupware systems, however, major focuses are still on general purpose middleware systems [9, 10]. Although there are research efforts on scientific workflow systems and biological research processes, they do not address those issues specifically involved in quantum chemistry [11–13].

For these reasons, workflow-based web portal systems customized for quantum mechanical calculation and designed to support computational Grids must be needed in order to make computational chemistry widely-applicable and effective for various nano/bio research work.

3 WISE

We have developed a Workflow-based grId portal for problem Solving Environment (WISE) which has the feature of integrating workflow, Grid and web

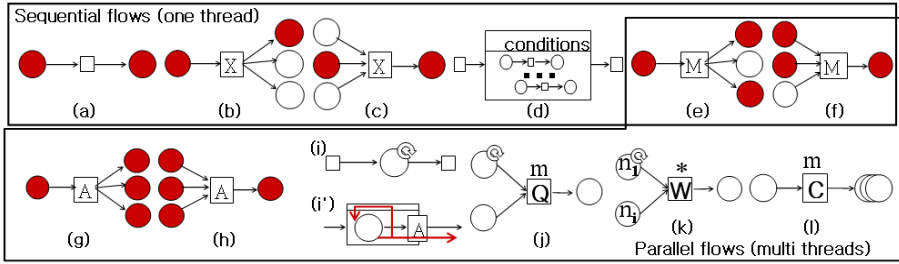


Fig. 1. Advanced basic control patterns; (a) Sequence (b) XOR-Split (c) XOR-Join (d) Loop (e) Multi-Split (f) Multi-Join (g) AND-Split (h) AND-Join (i) AND-Loop symbol (i') AND-Loop description (j) Queue (k) Wait (l) Node copy

technology to provide an enhanced powerful approach for problem solving environment. WISE provides new workflow patterns and GWDL which can overcome the limitations of the previous approaches by providing several powerful workflow patterns used efficiently to represent parallelisms inherent in parallel and distributed applications: pipe-line, data parallelism, and synchronization. To describe a workflow of grid application without ambiguity, we have proposed formally new advanced basic patterns such as And-Loop, queue, wait, node copy and etc. by classifying them into three categories; sequential, parallel, and mixed flow (See Fig. 1).

Our workflow-based Grid portal have been designed to provide a powerful problem solving environment by supporting a unified and consistent window to Grid which enables a substantial increases in user ability to solve problems that depend on use of large-scale heterogeneous resources. It provides users with a uniform and easy to use GUI for various interactive operations for PSE such as login/out, job submission, information search, file browsing, file transfer, and user profile, and especially supports interfaces for using workflow-based parallel programming environment on Grid, by supporting graphical workflow editor, resource finding, authentication, execution, monitoring, and steering.

WISE has a multi-layer architecture which can provide modularity and extensibility by each layer interacting with each other using the uniform interfaces (See Fig. 2). Also, we provide Model-View-Controller (MVC) design pattern that provides flexibility and modularity by separating the application engine control and presentation from the application logic for Grid services, and commodity-to-Grid technology for Grid service interface that supports various platforms and environments by mapping Grid functionality into a commodity distributed computing components.

4 QMWISE

In this section, we describe the architecture and user interface of workflow-based Grid portal for quantum mechanics.

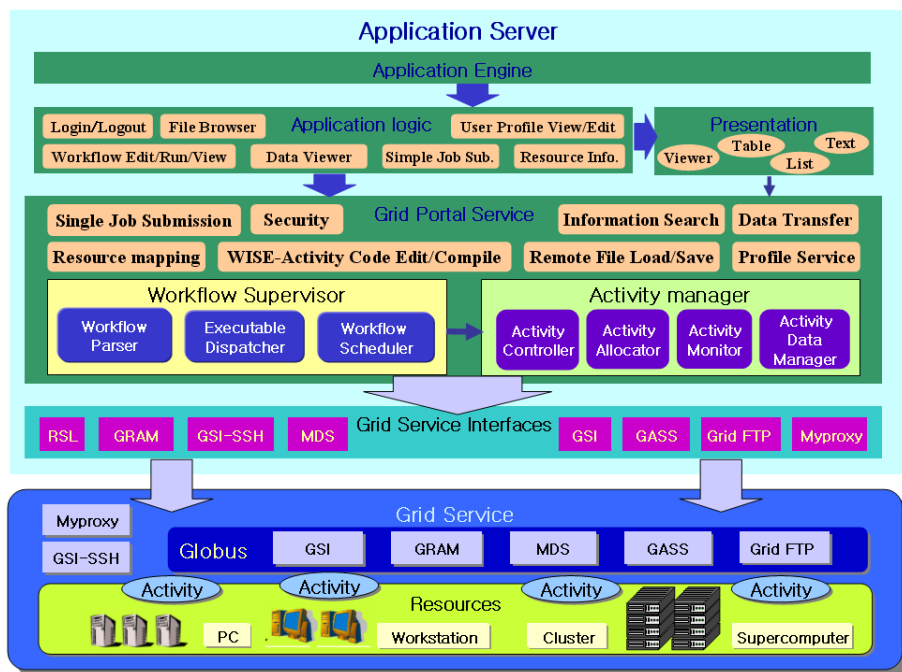


Fig. 2. The Overall Architecture of WISE

4.1 Architecture

For QM calculation, we add some modules to current WISE system in order to provide a convenient environment.

Input Genenerator. The Input Generator translates an input from user's web browser to XML, sends the XML to the Data Broker. An input from user includes information about the calculation to be performed, such as the job type, wavefunction type, the basis set, and symmetry information.

Data Broker. The Data Broker manages input and output XML data of a calculation process. The Data Broker communicates XML data with WISE system directly, and has information of the location of XML data. So when other modules need the specific data, they can get it through the Data Broker from many resources hidden by WISE system. For example, when the process is running, the Data Viewer fetches update data, and when the process is stopped or finished, the Controller fetches the result data through the Data Broker.

Workflow Manager. The Workflow Manager fetches the translated XML data from the Data Broker, generates workflow routines according to the user's input, submits the workflow to WISE system, and run it.

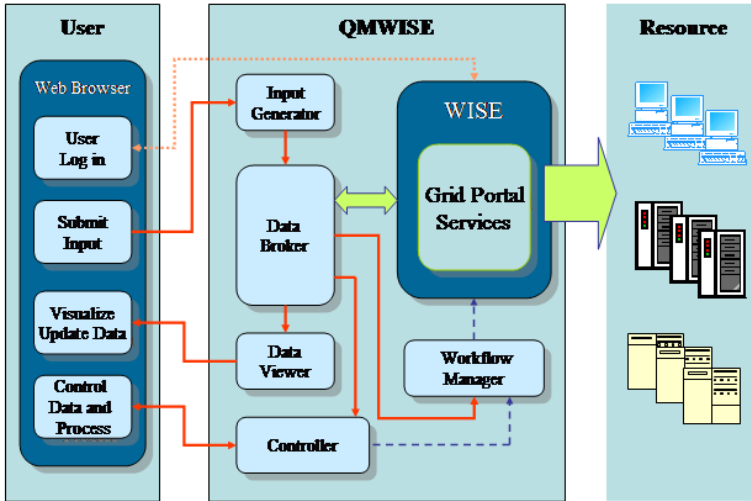


Fig. 3. The Architecture of QMWISE

Data Viewer. This module visualizes the calculation data while processing is running. The Data Viewer fetches the update XML data from each iteration calculation step through the Data Broker, interprets the data, and visualizes through the user's web browser.

Controller. The Controller has two main functions. The first is to stop the current process immediately when the update data shows that the process is against expectations. Also, The Controller can continue the previously stopped calculation by user's command. For using XML data format, there is no need to do further work like editing, converting the result in order to continue the calculation. The Controller makes the Workflow Manager to run it successively with previous data. The second is to provide a convenient environment to manage the result data through web browser. A user can seek and view the necessary data using the Controller.

4.2 User Interface

Our QMWISE provides the following functions which allow users to perform and control QM calculation easily.

User Authentication and Profile. A user is authenticated only once and provided all functions of our portal. QMWISE has user profile function which manages his/her information such as Grid certificate creation, update and management, list of available Grid resources, management of environment variables on many distributed host, re-request of resource authorization to its manager and email address.

Simple Input Submission. After a log-in, a user can submit the input value easily through his web browser. A user doesn't need to waste time to read the difficult and complicated manuals of QM calculation tools, only need the basic knowledge of the QM calculation such as the job type, wavefunction type, the basis set, and symmetry information. QMWISE provides an interactive web environment for input submission.

Visualization of Real-Time Update Data. While a calculation process is running, a user can see the real-time data of each step. Usually a calculation process takes a long time; a user must wait till the process is finished to know if the results are appropriate. It is a quite time consuming work. Our QMWISE provides the Data Viewer to show how the process is going.

Control Calculation Process. When the update data shows process goes wrong, a user can stop the process immediately; analyze the result data through the Controller. A user can submit a new input to web browser, otherwise continue the stopped calculation.

5 Conclusion

In this paper, we propose workflow-based portal for quantum mechanical calculations. Our portal uses Grid technologies provides advanced network services for large-scale, wide area, multi-institutional environments that require the coordinated use of multiple resources. Also, Grid technologies enable us to use unlimited resources for computing transparently. Providing web-based interface makes users compute quantum mechanical problems and approach quantum mechanical calculation programs in any place which is connected to Internet.

By translating user's input into XML, we can manage a lot of data efficiently in the Data Broker. As the Workflow Manager generates workflow routine automatically, non-experts who are not accustomed to QM calculation can perform the calculations. The Data Viewer shows the update data in real time, thus, users can know the state of the process. Also, users can control the process through the Controller.

Now QMWISE is being implemented and dependent on GAMESS. In the future work, we have a plan to extend our portal to including Gaussian, NWChem, in addition, MOPAC package for semi-empirical. Furthermore, our wish is to make our portal include molecular mechanics and dynamics simulation tools.

References

1. M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, J.A. Montgomery, "General Atomic and Molecular Electronic Structure System", *J. Comput. Chem.*, 14, pp. 1347-1363, 1993
<http://www.msg.ameslab.gov/GAMESS/GAMESS.html>

2. M. J. Frish, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montonmery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cislawski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chem, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. DeFrees, J. Baker, J. P. Stewart, M. HeadGordon, C. Gonzalez, J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1998, Gaussian Inc.
3. High performance Computational Chemistry Group, NWChem, A Computaional Chemistry Package for parallel Computers, version 4.1 (2002), Pacific Nothwest National Laboratory, Richland, Washington 99352, USA.
<http://www.emsl.pnl.gov:2080/docs/nwchem/nwchem.html>
4. J. J. P. Stewart, MOPAC93. Fujitsu Limited, Tokyo, Japan 1993
5. K. K. Baldridge, J. P. Greenberg, "Management of Web and Associated Grid Technologies for Quantum Chemistry Computation", ICCS 2003, LNCS 2660, pp. 111-121, 2003
6. T. Nishikawa, U. Nagashima, S. Sekiguchi, "Design and Implementation of Intelligent Scheduler for Gaussian Portal on Quantum Chemistry Grid", ICCS 2003, LNCS 2659, pp. 244-253, 2003
<http://www.gaussian.com/>
7. H. James, K. Hawick, P. Coddington, "An Environment for workflow applications on wide-area deistributed suystms", Hawaii International Conference on System Sciences, 2001.
8. G. Riempp, "Wide-Area Workflow Management", Springer-Verlag, 1998.
9. I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid Enabling Scalable Virtual Organazations", International J. Supercomputer Applications, Vol 15, No3, 2001.
10. I. Foster, C. Kesselman, J. Nick, S. Turecke, "The Physiology of the Grid : An Open Grid Services, Architecture for distributed system integration", Jan 2002.
11. G. Chin, K. Schuchardt, J. Myers, D. Participatory, "Workflow Analysis: Unveiling Scentific Research Processes with Scientists", Proc. Of the 6th Biennial Participatory Design Conference, 2000.
12. K. Kochut, J. Arnold, A. Sheth, J. Miller, E. Kraemer, B. Arpinar, J. Cardoso, "IntelliGEN : A Distributed Workflow System for discovering Protein-Protein Interactions", International Journal on Distributed and Parallel Database, Special Issue on Bioinformatics, 2002
13. M. Peleg, I. Yeh, R. Altman "Modeling biological processes using Workflow and Petri Net models, Bioinformatics", vol 18. pp. 825-837, 2002.

Agent-Oriented Formal Specification of Web Services

Hong Zhu¹, Bin Zhou², Xinjun Mao², Lijun Shan², and David Duce¹

¹ Department of Computing, Oxford Brookes University, Oxford OX33 1HX, UK
hzhu@brookes.ac.uk

² Dept. of Computer Science, National Univ. of Defense Tech., Changsha, 410073, China

Abstract. Web services (WS) provide a technology for integrating applications over the Internet. The components of a WS are active and persistent computational entities that have autonomous and social behaviours. The paper investigates the formal specification of WS architecture and applications within a caste-centric framework of multi-agent systems. An abstract specification of the general architecture of WS and an example of WS application are given in the SLABS language, which was designed for developing agent-based systems.

1 Introduction

As a distributed computing technology, Web services (WS) offer a promising approach to integrate applications over the Internet [1]. It is characterised by the dominance of program-to-program business-to-business interactions [2], hence widely recognised to be fundamentally different from existing distributed computing techniques.

The development of WS applications is bound to be complex and difficult for two main reasons. First, WS technology enables dynamic software integration at application level. Program-to-program interaction established at runtime implies that it may be impossible to determine the scope of integration at design time. There is little theory and practice of such integration in the software engineering literature. Second, business-to-business interaction implies that the integration can be within an enterprise as well as between enterprises. Thus, the software components in a WS application are usually developed by different vendors. The lack of communications between component providers and component users has long been recognised as a main cause of difficulties in component technology, but no satisfactory solution has been found. In the context of WS, recently, it is realised that, in addition to the descriptions of the syntactical aspects such as the formats of the messages, the description of semantic aspects such as business logic are of vital importance for the success of WS technology [3, 4]. Proposed solutions in the literature rely on ontologies for taxonomic descriptions of the functionality of each service, and on workflow for the restrictions on the orders that services are called [5, 6]. It is still unclear whether ontology and workflow descriptions are adequate to provide the required semantic information.

In this paper, we propose an approach that uses formal specifications to describe the semantic aspects of WS based on our caste-centric framework of multi-agent systems (MAS). We demonstrate the uses of an agent-oriented formal specification language SLABS [7, 8] to bridge the gulf between service providers and requesters.

2 Web Services as MAS

Agency is a fundamental concept in agent-based computing though what agenthood means exactly is a matter of controversy. People tend to define the concept by certain characteristic properties [9, 10]. Among many such properties, autonomy, proactivity, responsiveness and social ability have been widely considered as the most important. These properties match the features of software systems that constitute a WS application. The components of a WS application can be considered as software agents. For example, each provider or requester is autonomous. It can say 'go' to initiate actions such as to request for services. It can also say 'no' so as to refuse a service request. These components have certain social ability because of their dynamic discovery and invocation of services. At this level of abstraction, it is apparent that agent technology is suitable for the development of WS applications.

However, not all agent models are suitable for the development of WS. For example, BDI models define agents as computational entities with mental states that consist of belief, desire and intension [11, 12]. In such models, agents' behaviours are controlled by such mental states. Game theory models define agents as computational entities that aim to maximise their utility functions. WS has been considered as an attractive technology for wrapping existing applications and IT assets so that new solutions can be deployed quickly and recomposed to address new opportunities [2]. Few of existing IT assets can be considered as agents in these models.

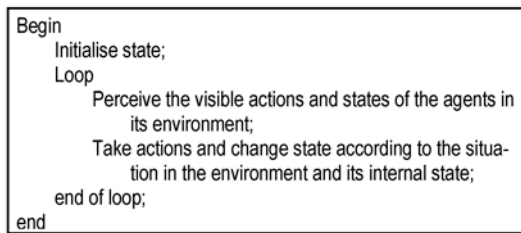


Fig. 1. The control structure of agent's body

Therefore, this paper take a software engineering approach to the analysis, modelling and design of MAS [13]. We define agents as active and persistent computational entities that encapsulate data, operations and behaviours and situate in their designated environments. Here, data represents an agent's state. Operations are the actions that an agent can take. Behaviours are rules that govern the agent's state changes and actions. By encapsulation, we mean that an agent's state can only be changed by the agent itself. In our model, agents' structure consists of a name, an environment description, a list of state space and action declarations, and a body in the form of Fig. 1 that determines its behaviour.

The central concept of our approach is *caste*, which is the classifier of agents. It is a new concept introduced by SLABS. In our model, the agents in a MAS are grouped into castes. The agents in the same caste have a set of common structural and behavioural characteristics. An example of behaviour characteristics is that an agent follows a specific communication protocol to communicate with other agents. The relationship between agents and castes is similar to that between objects and classes. The

difference is that an agent can join a caste and retreat from a caste dynamically at run-time. Inheritance relationships can also be defined between castes. A sub-caste inherits the structure and behaviour features from its super-castes. However, a sub-caste cannot override the structure and behaviour rules of a super-caste, although it can have some additional state variables, actions and behaviour rules. The parameters of the super-castes may also be instantiated in a sub-caste. The caste facility provides a powerful vehicle to describe the normality of a society of agents. Multiple inheritances are allowed to enable an agent to belong to more than one society and play more than one role in the system at the same time. Castes plays a central role in our methodology of agent-oriented software development [13, 14]. It distinguishes our approach from the others. In the SLABS language, castes are specified in the form shown in Fig. 2.

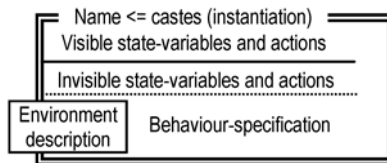


Fig. 2. SLABS's specification of castes

The components of a WS application can be modelled as agents defined above. They are divided into castes of service providers and service requesters. Different types of service requesters can also be further grouped into sub-castes so that components representing different types of service requesters are divided into the different sub-castes and have different structural and behavioural features. An agent can join a sub-caste to become a valid requester and retreats from the caste after the service is finished or when it is unsatisfied with the service. When it is a member of the caste, it must obey the behaviour rules in order to obtain the required services. But, it has no obligations to follow the rules after it retreats from the caste.

Agents are situated in their *designated* environments. By designated environment, we mean that the environment of an agent contains a specified subset of the entities in the system. This subset may vary at run-time within a specified range. In SLABS, an environment description specifies a collection of castes and a set of particular agents. A designated environment differs from a completely *open environment*, where every element in the system can always affect the behaviour of an agent. It also differs from a *fixed environment*, where an agent can only be affected by a fix set of entities in the environment. In both fixed and open environments, the agent cannot change its environment. It is worth noting that both fixed and open environments are special cases of the designated environments.

3 Specification of WS Architecture

The architecture of WS covers three main aspects of distributed computing: (a) a framework of the organisation of the software systems for access through a network; (b) the mechanism and facility for the publication and registration of the services so

that the services can be dynamically discovered; (c) a set of standards that enables components to exchange data with each other. In particular, the provided services are described in WSDL using a standard formal XML notation that provides all of the details necessary to interact with the service including message format, transport protocol and location. The services are published with a service registry that complies with a standard called UDDI. Once a WS is published, a service requester can find the service via the UDDI interface. Standards like HTTP, SOAP and XML are used for transportation and marshalling of parameters so that platform and language-independent access to WS can be achieved.

At an abstraction level above the technical details, the architecture of WS consists of three types of components: service registry, service providers, and service requesters. These agents belong to three different castes specified below.

The caste in Fig. 3 specifies service providers. It states that a service provider can have two actions: to register and unregister at a service registry. It has a visible state that describes its services. Its behaviour is specified by two rules: one for register and the other for unregister.



Fig. 3. Specification of service provider

The caste in Fig. 4 specifies service requesters. A service requester can make search requests to a service registry, but there is no restriction on when and what to search for. Therefore, there is no behaviour rule in the body of the caste.

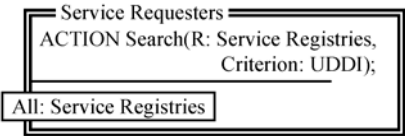


Fig. 4. Specification of service requester

Fig. 5 is the specification of service registries in SLABS. There are three rules for the behaviour of a service registry. The first states that when a service requester searches for a WS with a criterion, the registry must reply with a set of registered WS that matches the criterion. Here, we leave the function *Match* as a predefined function. The second and third rules deal with registration and unregistration, respectively.

Notice that, first, the semantics of SLABS implies that an agent can be a member of one or more castes. For example, a service provider can also be a service requester of another service provider. Second, an agent can join a caste and retreat from a caste

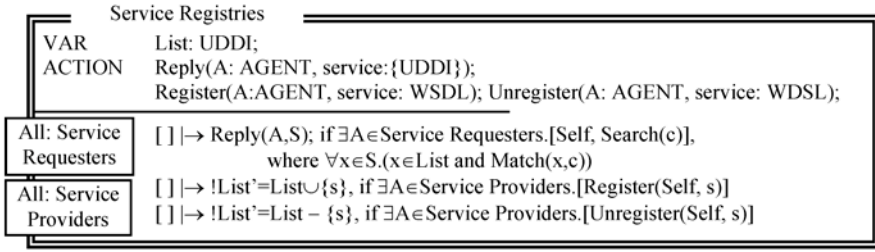


Fig. 5. Specification of service registries

at run-time. The membership relation is not static. Third, in the specification above, instead of giving all the details of the standards UDDI, SOAP and WSDL, we treat them as pre-defined data types and provide an abstract specification of the functionality and behaviour of the components. This enables us to focus on the logic of WS rather than syntactic and format details. Fourth, at the architectural level, there is no relationship between the service providers and service requesters. The interactions between them can be established at runtime and specified with the particular service provider and requester. Finally, the specifications given in this paper are for the illustration of the uses of SLABS. Some simplifications of the problems are made.

4 Specification of WS Service Providers

The specification of a service provider not only needs to define the services that it provides, but also the way that the services should be used. In a WS application, service requesters can be further classified into a number of types. Each of them can be specified by a caste.

For example, consider the online auction services. Two types of requesters may interact with an online auction WS. Sellers ask for the service provider to set up an online auction to sell its goods with certain conditions. Buyers can then bid for the goods online. Thus, we identify three different castes in this application: (a) Auction Service Providers, (b) Sellers, (c) Buyers. The caste in Fig. 6 specifies the behaviour of auction providers.

Auction Service Providers is a sub-caste of Service Providers. Sellers and Buyers castes in Fig. 7 and Fig. 8 are sub-castes of the Service Requesters caste.

The interactions between a service provider and a requester are often so complicated that an interaction protocol must be defined. In the online auction example, the protocol defines how to bid and who will be the winner, etc. It is defined by two sets of rules, one for the auctioneer and one for the buyers. The protocol specified in Fig. 3 is a simplified version of English auction. The rules restrict the behaviour of a buyer in an auction, but not on how individuals make decisions. Similarly, a protocol for the interaction between a seller and the auction service provider must be defined and specified. Details are omitted for the sake of space.

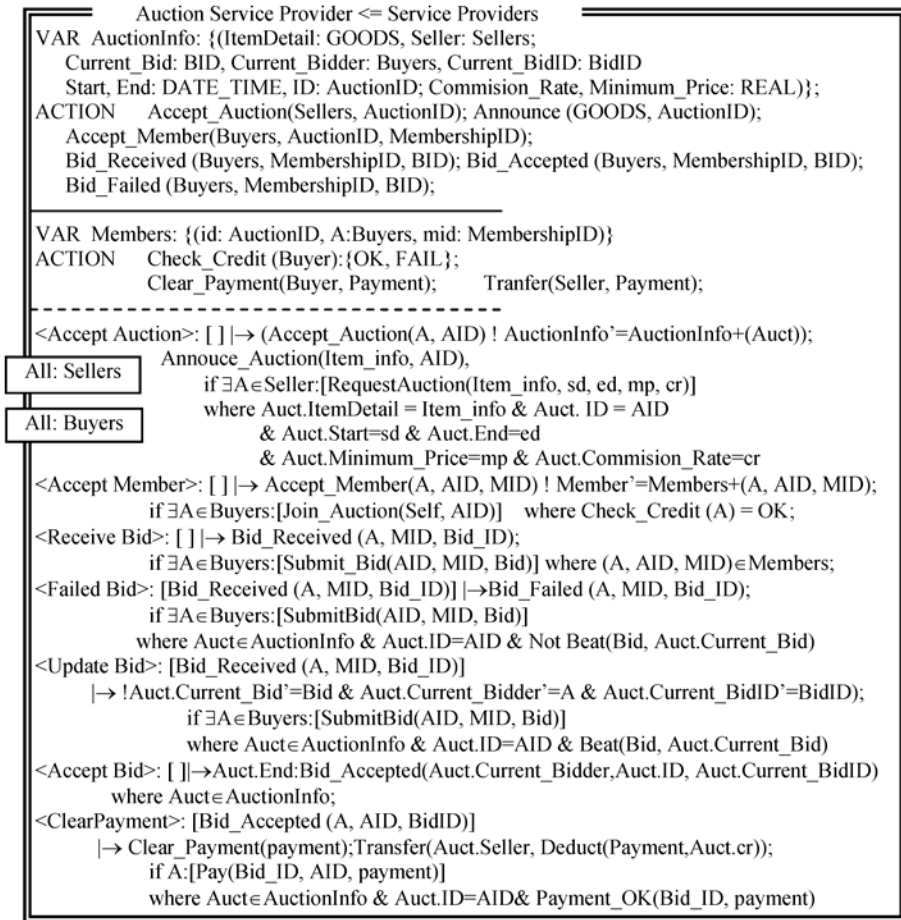


Fig. 6. Specification of auction service provider

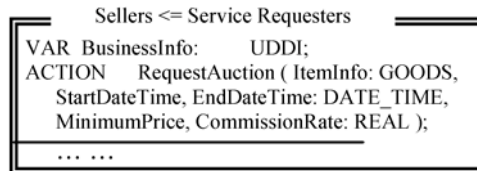


Fig. 7. Specification of seller

It is worth noting that in the above example a WS service provider is specified by one caste to define the provider's functionality and behaviour together with two castes to specify the expected behaviours of the service requesters. The specification of the requesters serves as the assumptions about the requesters' actions and behaviours. It explicitly states how the services should be used. The correctness of an implementation of a WS service provider can only be understood and proved by using all of these

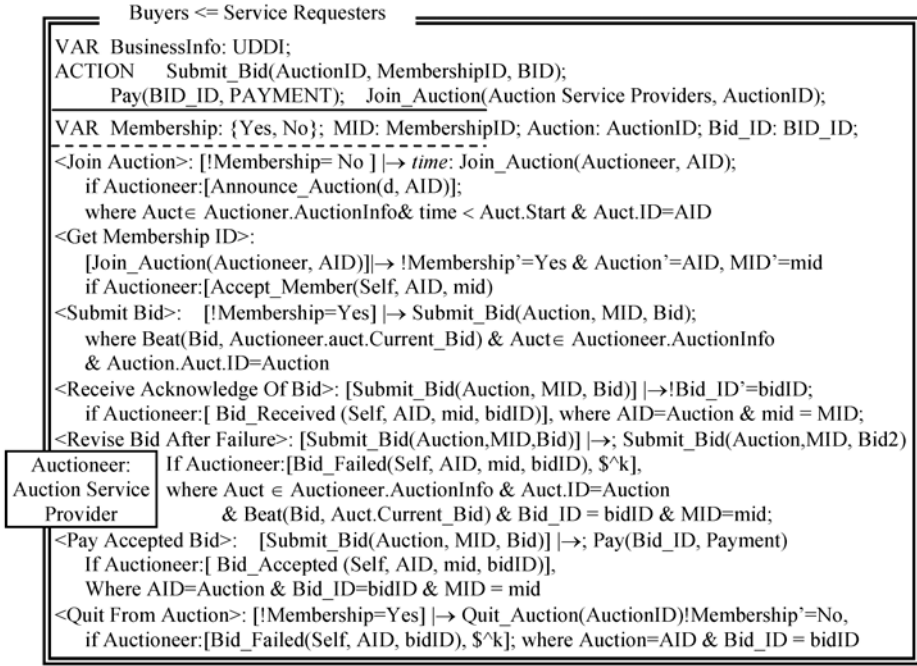


Fig. 8. Specification of buyer

castes. Such information is crucial for software developers not only on the service provider side but also on the service requester side. The specification of the requesters also leaves a great space of flexibility about their behaviour. For example, a specific buyer can have its own rules to determine when and what bid is to be submitted.

5 Specification of WS Service Requesters

To demonstrate how such a specification can be used for the development of requester side software, consider an online flight ticketing service that sells air tickets for an airline. Assume that, the specific application has a more concrete rule for deciding when to request online auction services. For example, the caste in Fig. 9 specifies a business rule that it will try to sell the unsold tickets by online auction when the time reaches 8 days before the scheduled flight.

The caste SellByAuction in Fig. 9 inherits the capability and behaviour of the caste Sellers for its interaction with auction service providers and a caste TicketSellers for its business rules. It also has an additional rule for its request of auction services. In general, the specification of business logic can be separated from the specification of the interaction protocol by using two or more castes.

An auction service requester may use a number of different auction service providers, say auctioneer A and B, to sell their products such as air tickets. In such a case, we can declare two agents as instances of the caste SellByAuction. Alternatively, agent A and B can be dynamically created as instances of the caste. Details of their specifications are omitted for the sake of space.

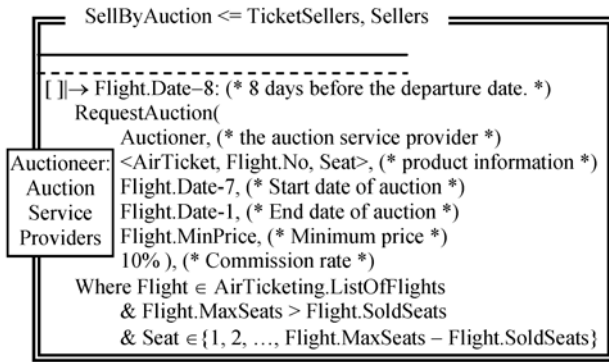


Fig. 9. Specification of air ticket seller who sells by auctions

6 Concluding Remarks

The approach to the formal specification of WS proposed in this paper can be summarised by two well-known software engineering principles. The first is the *principle of separation of concerns*. The specification of different kinds of components such as the providers and requester are separated into different castes. Different types of WS requesters and providers are further separated into sub-castes. The specification of private information such as business logic and internal decision making processes are separated from the specification of public information such as interaction protocols, communication protocols, etc. and specified in different castes. Such a modular structure of specification enables the application of the second principle, which is the *principle of information hiding*. The private information isolated in a caste can be hidden from public access. At the same time, the public information, especially the assumptions made by the service provider about the service requesters are specified. These principles are strongly supported by the caste facility. The specifications in SLABS are modular, composable and reusable.

There have been several efforts to define specification languages and/or standards for enabling software to use WS. Among the most well-known are IBM's WSFL [5] based on Petri Net theory, Microsoft's XLANG [6] rejuvenated the Pi-Calculus model, and BPMI.org's BPML 1.0 [15] that unified these two approaches. More recently, BEA, IBM, and Microsoft published BPEL4WS. Other organizations advocated radically different approaches for business process modelling, such as DAML-S [16]. There are two most important differences between SLABS and the above. First, WSFL, BPML and DAML-S focus on the workflow management of multiple Web Services, i.e. the execution orders and transactional issues. SLABS can specify these issues as well as other semantic aspects of Web Service. Second, SLABS is on a more abstract level while the related works are on a more operational level.

There are a number of problems that need further research. We are investigating how formal specifications of WS can be represented in XML format to facilitate the dynamic search and integration of WS applications.

Acknowledgement

The work reported in this paper is partly supported by China National High Technology Research and Development Programme (863 programme) under the grant 2002AA116070.

References

1. Lau, C. and Ryman, A., Developing XML Web services with WebSphere Studio Application Developer. IBM SYSTEMS JOURNAL, 2002. 41(2): pp178-197.
2. Gottschalk, K., et al., Introduction to Web services architecture. IBM SYSTEMS JOURNAL, 2002. 41(2): pp170-177.
3. Leymann, F., Roller, D., and Schmidt, M.-T., Web services and business process management. IBM SYSTEMS JOURNAL, 2002. 41(2): pp198-211.
4. Lambros, P., Schmidt, M.-T., and Zentner, C., Combine Business Process Management Technology and Business Services to Implement Complex Web Services, IBM Corp, 2001.
5. Leymann, F., Web Services Flow Language, IBM Corporation, 2001.
6. Thatte, S., XLANG-Web Services for Business Process Design, Microsoft Corp., 2001.
7. Zhu, H., SLABS: A Formal Specification Language for Agent-Based Systems. Int. J. of Software Engineering and Knowledge Engineering, 2001. 11(5): pp529-558.
8. Zhu, H., A Formal Specification Language for Agent-Oriented Software Engineering, Department of Computing, Oxford Brookes University, 2002.
9. Jennings, N.R., On agent-based software engineering. Artificial Intelligence, 2000. 117: pp277-296.
10. Lange, D.B. Mobile Objects and mobile agents: The future of distributed computing? in Proc. of Proceedings of The European Conference on Object-Oriented Programming, 1998.
11. Rao, A.S. and Georgreiff, M.P. Modeling Rational Agents within A BDI-Architecture. in Proc. of the Int. Conf. on Principles of Knowledge Rep. and Reasoning, 1991, pp473~484.
12. Wooldrighe, M., Reasoning About Rational Agents, 2000: The MIT Press.
13. Shan, L. and Zhu, H. CAMLE: A Caste-Centric Agent-Oriented Modelling Language and Environment. in Proc. of SELMAS'04 at ICSE'94, Edinburgh, UK., 2004, IEE, pp66-73.
14. Zhu, H., The role of caste in formal specification of MAS, in Proc. of PRIMA'2001. LNCS, Vol. 2132, 2001: Springer: Taipei, Taiwan, pp1~15.
15. BPML.org, The BPML specification version 1.0, <http://www.bpmi.org>.
16. Daml.org, The DAML Services Coalition. DAML-S: A Semantic Markup For Web Services, <http://www.daml.org/services/daml-s/2001/10/daml-s.pdf>.

Autonomic Incident Manager for Enterprise Applications

Renuka Sindhgatta, Swaminathan Natarajan, Krishnakumar Pooloth,
Colin Pinto, and N.S. Nagaraj

Infosys Technologies Limited, Electronics City, Hosur Road, Bangalore, India
{renukasr,swaminathan_n01,krishnakumarp,
colin_pinto,nagarns}@infosys.com
<http://www.infosys.com>

Abstract. Enterprises have been facing the challenge of reducing their IT Operation and Maintenance costs to provide greater Return on Investment (ROI). The emergence of standards and guidelines has enabled efficient tracking and management of the activities required in this space. While standards have been useful in ensuring process completeness, all the tasks require to be performed manually by the operations personnel. There is possibility of improving the execution effectiveness by developing self managing support systems. This paper proposes the use of Knowledge Based Systems and Web Services for developing an Autonomic Incident Manager for Enterprise Applications to provide productivity benefits in a rather disregarded field of software maintenance and support.

1 Introduction

The increased dependency of Enterprises on IT has resulted in the need for constant support and maintenance of the IT systems and applications. Consistent and effective post-deployment support of IT systems and applications plays a key factor in determining the effective value IT provides to the enterprises. This criticality has led to emergence and adoption of IT Infrastructure Library (ITIL) [1] – a standard that defines the process with best practices and activities for IT support services. The process however, is manual and requires application experts to manage and solve glitches in the systems (also known as incidents).

Incidents are events that are not part of the standard operation of an IT system and reduce its quality of service. Abort of a Job due to unavailability of a file can be one such example of an incident. Incident management is a well defined process that defines the activities to be executed to ensure smooth handling of incidents and restoration of IT systems to normal operation. The activities defined in the incident management process include:

- Detection of the Incident
- Classification of the Type of Incident
- Investigation and Diagnosis of Incident
- Resolution and Recovery of the Incident
- Closure of the Incident.

Incident Management is an important task performed by an operations team any enterprise. There are several COTS (Commercially Available Off-the-Shelf) tools that support tracking and management of the activities involved. However, the efficiency of an incident management team is largely dependent on the availability of experts. The paper presents an Autonomic Incident Manager (AIM) capable of handling all the activities related to incident management in an autonomic manner. AIM comprises of two key technology elements - knowledge based system and Web Service. The paper focuses on the architecture and process of developing and deploying AIM. It is based on the experience and insights gained while working with a team supporting IT systems of a major financial enterprise. An experimental set up was created to verify the feasibility of building an AIM.

1.1 Motivation

IT systems and applications have formed the backbone of businesses of most enterprises for over decades. Incidents that occur in these systems are handled by a support team that ensures all incidents are tracked and managed and the businesses run smoothly. The support team has various levels of support – first line, second line and the third line support. The first line and the second line support teams usually handle incidents that are repetitive – same incidents occur again and again. In the first line support, the team follows a set of procedures to resolve the incident. In the second line support some diagnosis is done to identify the cause and provide a solution. An example of such an incident would be - failure in upload of a sales order document in a B2B application. The diagnosis would require step by step analysis of the various documents and acknowledgements that were transferred as a part of the process and the identification of the error in the documents. The third line of support deals with problems related to lower level components of the applications. An error in the oracle database is one such incident. The incidents handled by third line support are unique and require an expert to resolve. 85% of the incidents are handled by the first and the second line support. As these incidents are repetitive, a standard set of conditions can be associated to the incident based on which a resolution can be applied. Hence, these are viable for self monitoring and analysis.

IT systems and applications grow over time, re-engineering their architecture is often seen as a task with humongous effort and risk. Making any drastic change to the already existing systems is difficult. Hence, a mechanism of providing a system that works with the existing enterprise application to provide self-healing capabilities is the most feasible alternative. Currently, Autonomic computing has been confined to centralized systems [4]. AIM, however, focuses on enterprise applications that comprises of multiple executables, servers, databases or resources. Building a self-managing and monitoring system in a distributed environment requires multiple technology elements.

The following sections describe the architecture of Autonomic Incident Manager. Currently, it can be realized that 15% of the incidents are still not in the scope of the AIM. However, addressing majority of the incidents would provide substantial benefits to the enterprises.

2 Autonomic Incident Manager (AIM)

AIM aims to automate the first and second line incidents for any enterprise application. It primarily consists of a monitoring system and an inference engine. The inference engine diagnoses the incident by the querying the status of the production environment— status of the files, servers, jobs , etc. Based on the information obtained, the root cause is identified. Once, the diagnosis is complete, the required resolution is executed on the production system. The inference engine is a knowledge based system consisting of the domain and diagnostic knowledge of enterprise applications. The other critical components that AIM are shown in figure 1. The details of each of them are described in this section.

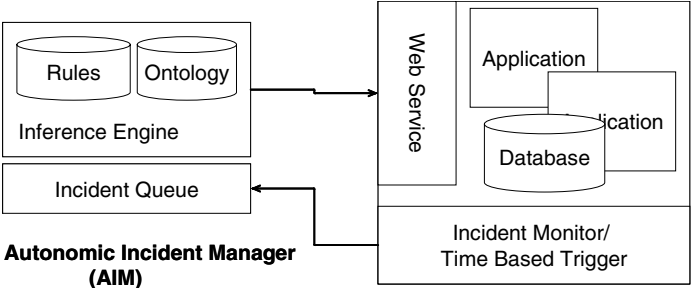


Fig. 1. Autonomic Incident Manager

1. System Knowledge Definition and Acquisition
2. Diagnosis Knowledge Definition
3. Web Service Definition and implementation
4. Incident Monitor and Trigger.

2.1 System Knowledge Definition and Acquisition

This phase involves the identification of knowledge items. Knowledge items are the concepts relevant to the application domain and limited by the scope of the incidents, the technology and the application architecture.

The knowledge items are classified into different categories – infrastructure, support application, support policy and support organization. Support System ontology is developed. Sample system ontology developed using Protégé 2000 is shown in figure 2.

In the given example, *Support application* consists the jobs, the files, databases and any concept related to the enterprise application execution. *Infrastructure* consists of the resources related to the application infrastructure like hardware systems and servers. The *support policy* contains the contextual inputs related to application execution policies and support policies like calendar dates for batch jobs to run, the service levels for incident diagnosis, the business policies impacting support. Finally, the *Support organization* would contain the details of the support personnel and their reporting hierarchy.

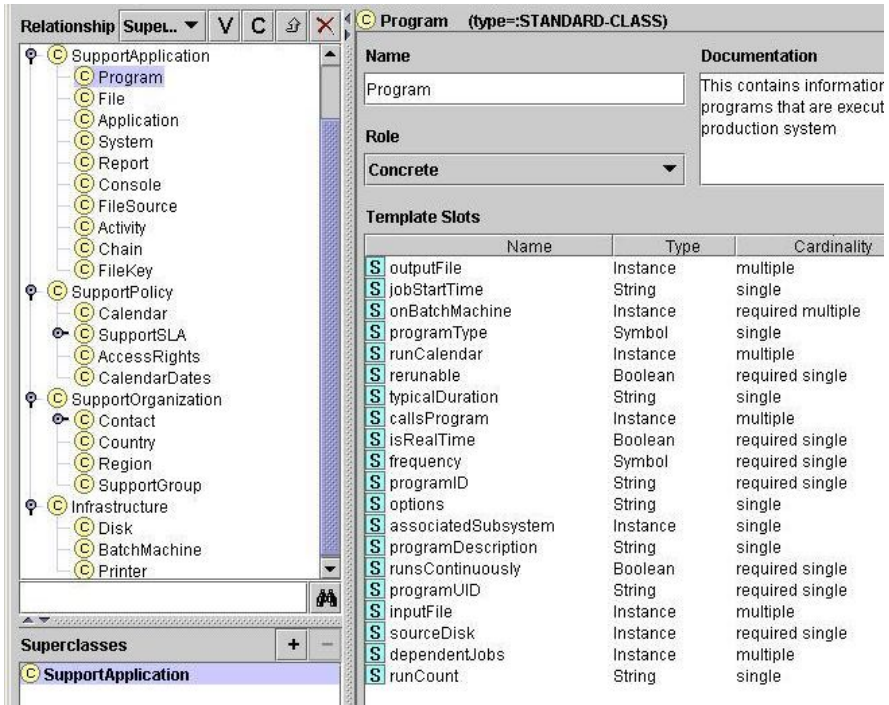


Fig. 2. Application Support Ontology

Once the ontology definition is complete, the instance information for each knowledge item is captured. This knowledge has to be extracted from several sources. There are Commercially Off the Shelf (COTS) tools[3] that enable application mining giving relevant data. In the absence of these, relevant information can be extracted from job schedulers, application code, and documents. The ontology with instance information is shown in figure 3.

2.2 Diagnosis/Inference Knowledge Definition

Commonly, for first and second line incidents, the diagnosis requires a series of analysis on the resource dependencies as well as their runtime status – status of jobs, error in the log file, etc. The process of diagnosis follows by building inferences based on the runtime status that finally lead to problem identification and resolution. Use of the inference knowledge along with domain knowledge has been described in CommonKADS [6] for building Knowledge Based Systems (KBS). A Rule Base is built by identifying the steps taken by an expert in resolving incidents. The rules use three distinct ontology elements for diagnosis.

Incident Information/Alert: It defines the details of the incidents. It contains the information of the incident that has occurred. For example a Job Failure would be defined as JobFailedAlert in the Ontology containing jobName and failureTime as slots.

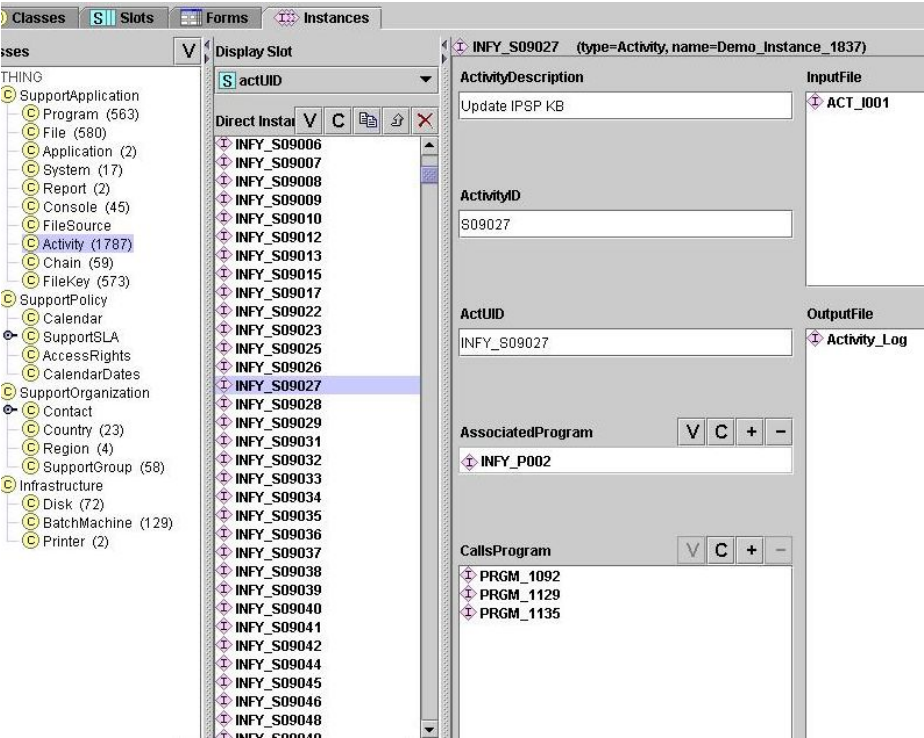


Fig. 3. Ontology Instance Information

Static System Information: As described in the earlier section contains the resources and context information of the enterprise system with details of the dependencies.

System Status Information: The status information is also captured in the ontology. For each resource there is an associated status at run time. The details that need to be captured as a part of diagnosis is defined in the ontology. A FileStatus would contain the status of a file having filename and fileStatus as slots.

The expert defines the diagnosis knowledge as rules. A typical diagnosis consists of identifying the symptoms that further lead to identification of the root cause. Given the root cause, the problem can be solved. Hence, there are two types of rules that are defined by the expert – to identify the symptoms, to resolve the problem.

- Status Query Rules – Status Query rule queries the status of the resource based on certain conditions. Hence, the action of a status query rule results in querying the status of a resource.
- Resolution Rules – The rule provides the resolution by checking the status of various resources.

An example of simple rules using CLIPS Rule engine is given figure 4. CLIPS has been shown due to its support for frame based knowledge bases with COOL (CLIPS Object oriented language).

There can also be inference rules that monitor the health of the systems. Scenarios where a set of job completions, file transfers and database connections need to be checked at certain time intervals can be codified in the rules. By defining a time period alert that triggers the query rules, the health of the system can be checked. The advantage of such an approach is the use of the system knowledge already available in the repository - Dependent files, Dependent Jobs, databases, etc.

```
; A Query Status Rule
(defrule AlertArrived
  ?ProgramAlertObj <- (object (is-a JobFailedAlert))
  ?ProgramObj <- (object (is-a Program))
  (test( eq (send ?ProgramAlertObj get-programID)
            (send ?ProgramObj get-programID) ))
  =>
  (getProgramStatus (send ?ProgramObj get-programID)))
)
; A Resolution Rule
(defrule AlertArrived
  ?ProgramAlertObj <- (object (is-a JobFailedAlert))
  ?ProgramObj <- (object (is-a Program))
  (test( eq (send ?ProgramAlertObj get-programID)
            (send ?ProgramObj get-programID) ))
  (test( eq (send ?ProgramAlertObj get-programStatus)
            ON_HOLD))
  =>
  (CheckAndRestart (send ?ProgramObj get-programID))
)
```

Fig. 4. Query and Resolution Rules

2.3 Web Service Definition and Implementation

The choice of web service is made as it is platform independent. An application can comprise of several components deployed on multiple platforms and hence, the incident manager should be capable of querying the status and placing actions on all the platforms on which the application is deployed. A web service that implements all the status queries and actions on the given platform is required. Hence, on the system where the programs executes, a web service that enables status query of programs or files (query service) as well as the web service capable of restarting the job is available (resolution service) is required. The inference engine will be a web service client calling the query or resolution web service as exposed on the production systems. The detail of the server to be connected is picked from the ontology and is a part of the rule. Use of Web Service provides advantage of working in a distributed environment of servers.

In building such a system, a framework was designed to ensure easy addition of query and resolution services. On the CLIPS engine side, an XML file containing the details of query and action containing the function name and parameters are sent to the web service. The Web service parses the XML and calls the required implementation of the services on the production server. Thus, addition of a query or resolution

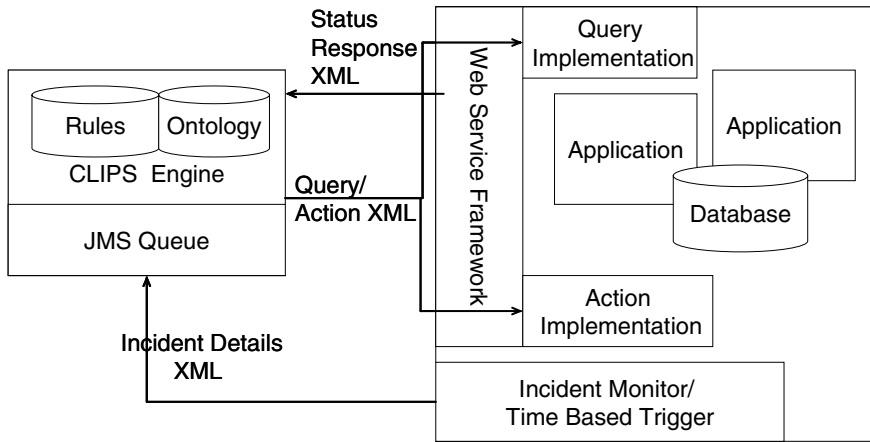


Fig. 5. Details of messages transferred between AIM and Production Systems

service would require implementation of the function on the production server. The call, translation and parsing was taken care of by the framework. A detailed view of AIM is shown in Figure 5.

2.4 Incident Monitor and Trigger

As an autonomic system should be capable of identifying the incidents, an incident monitor is required to check the health of the system and trigger the inference engine on occurrence of a glitch in the normal functioning of the system. A failure in the system can be identified by several tools – Job Schedulers, Service Desks, User mails, etc. This component should interface with these tools and on failure should send a message to the incident queue. The message is picked up on a sequential basis from the queue and is translated to a trigger in the inference engine and which is diagnosed. Similarly, a health check could be initiated by triggering a dummy alert at regular time interval from the Incident Monitor. This will execute the associated rules in the rule engine capable of doing a system status health check by calling the query services. On identifying failure, a failure alert could be sent to the incident queue for further processing of the incident.

3 Conclusion

There is a wide possibility of automation that can be brought in incident management. In this paper we present a mechanism of building an Autonomic Incident Manager using Knowledge Based System.

However, there are several challenges in deploying these systems as IT teams are typically sensitive to using or deploying new applications that work with their exist-

ing production systems. This is due to the critical impact IT systems have on their business. Acceptance to Autonomic system that requires deployment of new components on the existing system will take some more time and require experimental set ups and case studies. These systems will, however, help improve the execution efficiencies of support service – currently an imperative need of most enterprises.

References

1. Office of Government Commerce, ITIL Service Support, The Stationary Office of OGC, (Sep 2000)
2. Gartner Research Note, Legacy Understanding MQ: Knowledge for E-Business, M-12-8943 (2001)
3. Jeffery Undercoffer et al., A Target-Centric Ontology for Intrusion Detection, CADIP Research Symposium (2002)
4. LanFranchi et al., Towards a new landscape of systems management in an autonomic computing environment, IBM Systems Journal, vol. 42, no. 1, (2003) 119-128
5. S. Staab et al., Knowledge Processes and Ontologies, IEEE Intelligent Systems, vol.16, no. 1,(Jan./Feb. 2001) 26–34
6. Schreiber, G et al., CommonKADS: a comprehensive methodology for KBS development, IEEE Intelligent Systems, vol 9, no. 6, (Dec 1994)28-37
7. D. Fensel, Ontologies :A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, NewYork (2001)
8. R.Want et al., Comparing Autonomic and Proactive Computing, IBM Systems Journal, vol. 42, no. 1,(2003) 129-135

HiGAF: A Hierarchical Grid Accounting Framework

Qi Qian and Minglu Li

Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, P. R. China
{qian_qi, mlli}@sjtu.edu.cn

Abstract. Grid computing is a promising platform for executing large-scale resource intensive applications. Resource accounting is a basic and important activity in Grid research. In this paper¹, we propose an accounting framework based on hierarchical design. It comprises two layers of accounting managers: global accounting manager and local accounting manager. The latter provides the former a series of standard interfaces of accounting functionality by wrapping underlying legacy accounting systems. Thus different existing accounting systems are allowed to be reused and deployed into Grid environment with not too much effort.

1 Introduction

Grid computing is emerging as the next generation solution for sharing, utilizing and integrating resources among geographically distributed organizations or administrative domains. Resources connected via the Internet with middleware supporting remote execution of application constitute what is called “computational Grid” [1].

In a practical commercial and scientific Grid community, both the resources owners and users want to maximize their benefits. When user applications finish utilizing Grid resources, the resources consumed by the user applications should be accounted for and charged. So accounting and charging functionalities are indispensable to construct a feasible, robust and stable Grid [2].

We advance a hierarchical Grid accounting framework (HiGAF). In this framework, resource accounting is managed by a global accounting manager, which coordinates universal accounting functionalities and several local accounting managers, which manage the accounting at the organization level, potentially interfacing to organization-specific accounting and charging handling system. By separating local resource accounting operations from global resource accounting policy, we facilitate the complicated accounting tasks to a large extent. The two-level architecture and standard interface enable different local legacy accounting systems (LLAS) to join in the global accounting framework.

The rest of this paper is structured as follows. In the next section, we review current accounting solutions for Grid environment. Then we present our framework’s

¹ This paper is supported by 973 project (No.2002CB312002) of China, ChinaGrid Program of MOE of China, and grand project of the Science and Technology Commission of Shanghai Municipality (No. 03dz15027).

advantages and its detailed architecture in section 3. Four typical working scenarios of HiGAF are discussed in section 4. We summarize the paper and discuss future directions in the last section.

2 Related Works

Resource accounting has been researched in some Grid projects and organizations. [3] briefly discussed the challenges of correctly choosing and quantizing the items to be charged and described the scheme of an implementation based upon the concept of Home Location Register (HLR). It also tried to address the problem of local accounts management on Grid resources, proposing the use of a system of dynamically creating accounts called template accounts. But in its architecture, every resource has a HLR so the overall overhead will be very high in many cases.

[4] provided a secure Grid-wide accounting and payment handling system. It highlighted implementation issues with a detailed discussion on the format for various records/databases that the Gridbank needs to maintain. It also presented protocols for interaction between the Gridbank and other various components within Grid computing environment. But Gridbank didn't make good use of those LLASs and thus could not be expanded to provide multiple branches across the Grid to achieve scalability.

[5] introduced charging and accounting item in a grid computing system, and proposed a method for calculating the cost of a grid usage. Further more, it analyzed the demands of a charging and accounting system in a computational economy based grid. An architecture of charging and accounting system was designed in this paper. However, it is still a central-control system.

[6] defined the Computational Grid Service (CGS) which wrapped the Grid Service that was to be sold in the context of OGSA, and how it interacted with the Grid Banking Service (GBS). The document standardized the service data elements and service interface for the CGS and the GBS. But it did not offer an implementation. We think that future Grid tends to build on OGS and OGSA, so exposing accounting functionality as services is the most probable way to achieve universal acceptance and realization.

In summary, our review of current Grid accounting approaches revealed a range of valuable solutions, but they lack extensibility and flexibility to some extent. Our architecture makes good use of the LLASs in various organizations and eliminates the performance bottleneck in central accounting management by decomposing it into two hierarchies.

3 HiGAF Architecture

In this paper, we firstly review how an accounting system interacts with other components in computational Grid. After that, we introduce a new hierarchical Grid accounting framework (HiGAF) and list its advantages over its previous counterparts.

3.1 Interaction with Other Components

A Grid can be viewed as composed by three layers of participants: resource users, resource providers and system components. Accounting system belongs to the system components layer, and it interacts with other parts of Grid to perform its functionality. In an economy-based Grid environment, the whole Grid is driven by user and provider’s interests, where the accounting system has the following scenario (Fig. 1):

(1) Grid User Boker (GUB) queries Grid Accounting System (GAS) for resource prices to choose those matching its price policy. The detailed algorithms of price generation and resource selection are beyond this paper’s field.

(2) GAS analyzes user’s financial honor and deposit status to check whether user is permitted to execute his job using the resource. Possible underlying accounting support of debt may relaxes the strict requirement that user should have enough Grid Credit (GC) to run the job.

(3) When user application terminates, GAS performs corresponding activity according to the termination reason. If the job completes successfully, GAS tries to transfer resource usage expense from user’s account to provider’s account; sometimes it will append a debt record when user has spent more than his savings. If the application exits abnormally because of provider-side faults, GAS may make some compensation in accordance with provider’s policy.

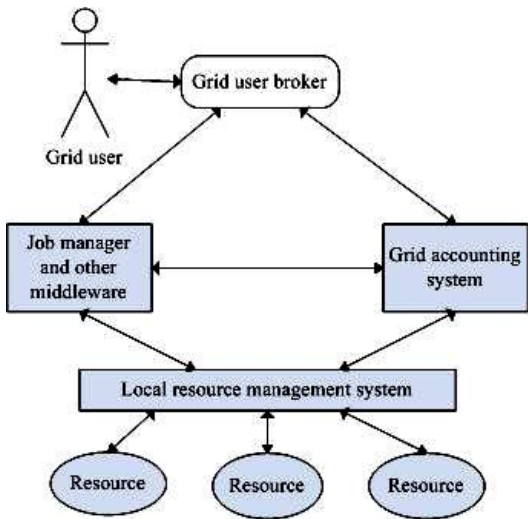


Fig. 1. Accounting System in the Grid

3.2 Hierarchical Architecture

HiGAF builds on the idea of decomposing accounting functionality and complexity. We break down the centralized accounting system into two hierarchies: Global Accounting Manager (GAM) and Local Accounting Manager (LAM), which altogether constitute the HiGAF. Their relation can be imagined as that between headquarter and branch.

As illustrated in Figure 2, GAM is located in central server in Grid and LAM exists in specific administrative organization. GAM plays the role of interacting with other middleware components in Grid and coordinating the universal payment issues among various LAMs, while LAM is responsible for communicating with LLAS. Once GAM determines the location of the resource that user will make use of, it can transfer most workload to that LAM. In this way we are able to decrease the degree of complication and bottleneck previously existing in a central-control accounting architecture. The lower-level interfaces exposed by LAM to GAM are standard and extensible, allowing for various new LLASs to be easily and seamlessly integrated into HiGAF.

In the following part, we will give detailed description on every component and their functions:

Price Repository (PR) – Global PR (GPR) and Local PR (LPR) store the prices of resources in Grid. LPR queries underlying LLAS for price. The price in GPR is just the mirror of price in LPR, so one vital issue is how to keep price in GPR valid at any moment. Our solution is to update the price in GPR regularly, and to append every price record in GPR a valid timestamp. If the price information is out-of-date because of network failure in update process, GUB will ask GPR to update price from LPR immediately.

Global Payment Dispatcher (GPD) – GPD is the key component in HiGAF on system-level. It receives user's choice of resource, verifies that the user has enough financial support to run his application, dispatches the accounting task to the corresponding Local Payment Controller.

Grid Honor Repository (GHR) – It stores Grid user's financial honor. GPD always looks into it to confirm that the user owns good historical usage record to run application. That means, if the user owes a large amount of debt during past Grid usage, he is not entitled to further usage until he pays off all his debts. When user's application completes, GHR may be updated if user exceeded too much beyond his consumption capability.

Remote Accounting Collector (RAC) – Once user does not have enough credit to use the resource in one LAM, RAC will collect deposit status of the same user distributed in other LAMs to assist GPD to authorize user.

Gatekeeper – It is the common interface provided by LAM to GAM. All the interactions between GAM and LAM are via this component. This component can change private currency in one LAM to universal Grid Credit and vice versa. We assume this function necessary as chances are high that different LLASs use different local currencies. In addition, job exiting status is reported to Gatekeeper when job succeeds or fails, enabling LPC to perform a transaction or compensation.

Local Payment Controller (LPC) – LPC is used to control the accounting process in local administrative organization. It temporarily stores user's deposit status from other LAMs while the job is running, starts the transaction or compensation when job terminates, passes deposit status to LTE, and reports to Local Overdraft Manager (LOM) if debt happens after payment. Another function is to consult LLAS when RAC needs the user's deposit status across various LAMs and to pass it to Gatekeeper.

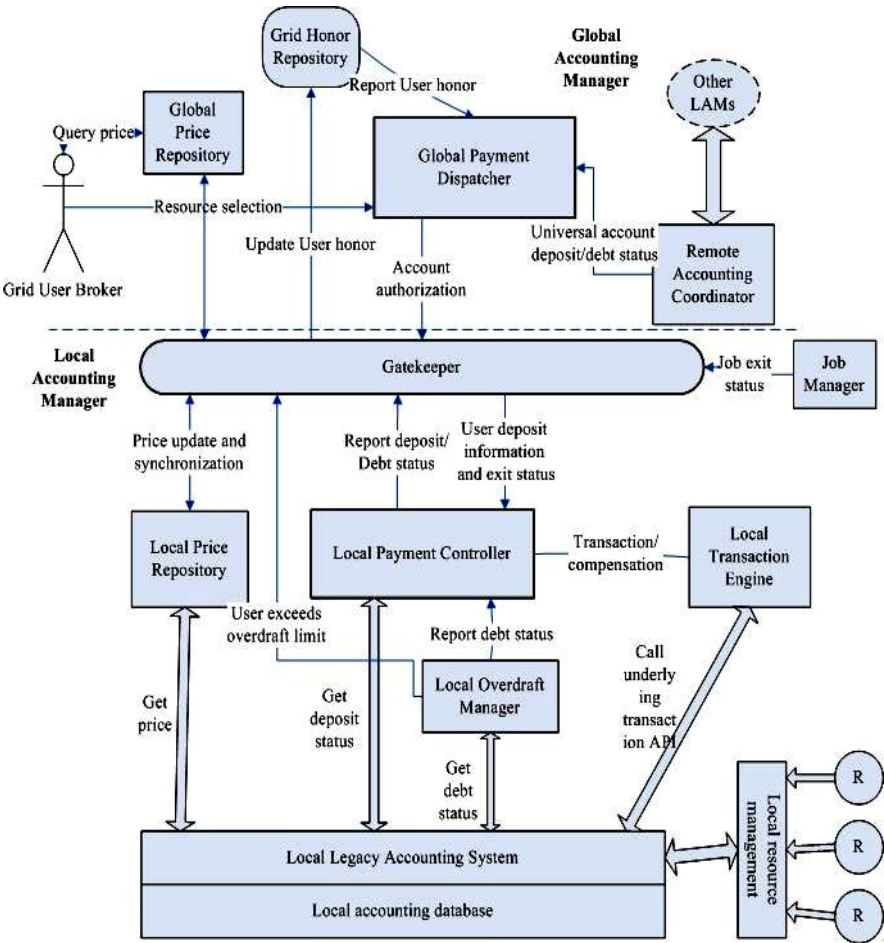


Fig. 2. Architecture of HiGAF

Local Overdraft Manager (LOM) – This component maintains the debt status of the user, and interacts with underlying overdraft manager. It reports to LPC when debt status is needed and updates GHR via Gatekeeper if user has exceeded debt limit.

Local Transaction Engine (LTE) – This is the actual component that is responsible for funds transfer between user account and provider account. It moves funds from one account to another, the direction depending on the type of the transaction – charging or compensation. The deposit status after transaction is reported to LPC for update in external LAMs, if necessary. It may not transfer real credit because user has already owed debts, in this case, debt status is passed to LOM.

3.3 Advantages

There have been some accounting systems developed and deployed in Grid computing environment [3, 4, 5]. Since they all focus on central-control system architecture,

flexibility, scalability and extensibility are not well supported by them. In addition, they tend to design a new accounting system for the whole Grid, thus wasting the previously built accounting capability. HiGAF has the following advantages over them:

- Investment protection: The chances are high that one virtual organization or administrative domain participating in the Grid has already set up its own accounting system. To force it to use the universal Grid accounting system instead of its own will undoubtedly waste the investment it has made and import many compatibility problems. Our framework will reuse the existing accounting infrastructure by wrapping an interface for it, whose overhead is very low.

- System-level accounting workload decomposition: For a Grid containing central accounting system, there is only one central accounting manager dealing with all payment-related transactions. So when the throughput is extremely high, as is very probable in real-world Grid, the accounting manager becomes the performance bottleneck and easily crashes down. Our design dispatches accounting tasks to local accounting systems.

- Allowing debts during usage: To simulate real-world trade process, we import “debt” into HiGAF, aiming at encouraging more resource usage. This issue is very new in Grid.

4 Working Scenarios

In this section we present four typical working scenarios of our accounting framework, in order to explain the mechanisms of economic transaction in the context of HiGAF.

We have to point out that the framework implementation is independent from the underling pricing algorithms and local accounting policies. The GAM only knows that there exist many LAMs in Grid, and the lower-level interfaces GAM calls are standard, although LAM is LLAS-specific.

We suppose that GUB has already enough information on user resource requirement and resource selection algorithm. We only focus on accounting functionality in Grid, thus ignoring other system-level middlewares by just mentioning their interaction with HiGAF when necessary.

We don't care how LLAS computes resource usage and expense, and how it performs charging and accounting. Our goal is to give clear explanation of how GAM cooperates with LAMs to account from a higher perspective.

- Scenario 1: Establishing accounting environment for user application locally.

Step 1. GPD checks in GHR for user's reputation to see whether he is entitled to using Grid according to previous usage records. If the user is infamous of not paying debts, the Grid will forbid the user from any application execution. The process terminates immediately.

Step 2. GUB queries resource prices from GPR. If GPR contains validated prices, it will respond them to GUB; otherwise, GPR has to synchronize with LAMs before responding.

Step 3. After GUB chooses the resource it prefers to using for job executing according to its algorithm, it submits its choice to GPD. We assume the resource is located in VO(i), but the case can be easily extended to include multiple resources distributed in different VOs.

Step 4. GPD verifies whether user has enough Grid Credit to run his application. It will firstly try to get the user's deposit information from the LAM(i). If user holds enough credit in LAM(i), the process of establishment ends. Or else it will go to scenario 2.

- Scenario 2: Establishing financial environment for user application globally.

Step 1. GPD asks whether user is willing to transfer his credit from other LAMs to the LAM where he is poor and his wanted resource is located. If he is willing, the process continues, or the whole job execution process terminates.

Step 2. GPD requests RAC for user's deposit status in other LAMs of Grid and accumulates them to check whether the sum is enough to use the resource. If it is enough, GPD tells LAM(i) to initialize an accounting process locally, passing information of how much deposit the user owns in every LAM for future transaction. The process completes. Or else it goes to step 3.

Step 3. If the sum is not enough for the resource yet, GPD consults RAC of how much overdraft the user can get from those LAMs. If the sum of deposits and overdrafts is more than job cost, the user is entitled to run his application, although leading to debts in several LAMs. GPD passes similar information as in step 4 of Scenario 1. The process completes. Or else it goes to step 4.

Step 4. The sum of user's deposits and overdrafts is less than the estimated job cost, the process terminates and the user is informed that he is not capable to run his job on the Grid.

- Scenario 3: Charging on successful job completion.

Step 1. Job manager notifies Gatekeeper(i) in LAM(i) at the time of job completing successfully.

Step 2. Gatekeeper(i) guides LPC(i) to perform a transaction between user account and provider account. LPC(i) passes user deposit information to LTE(i).

Step 3. LTE(i) tells LLAS(i) to transfer user's fund to provider's account LLAS(i). Note that this step may involve activities in external LAMs. LTE(i) decreases the corresponding deposits in those LAMs via Gatekeeper(i).

Step 4. LOM(i) may update the GHR if user has spent more his savings and list the user as disreputable. By doing so, the user is prohibited from further Grid usage until paying off the debts.

- Scenario 4: Compensating on job failure.

Step 1. Job manager notifies Gatekeeper(i) in LAM(i) at the time of job failure during execution.

Step 2. LPC(i) guides LTE(i) to compensate the user according to provider's policy.

5 Conclusion and Open Fields

In this paper, we present a hierarchical Grid accounting framework (HiGAF) from a high perspective. The traditional accounting infrastructure is decomposed into two layers: global accounting manager and local accounting manager. The local accounting manager is VO-specific and presents a standard interface to global accounting manager. By doing so, we can integrate various legacy accounting system into Grid easily. Four typical working scenarios are discussed within the context of HiGAF.

The potential application of HiGAF is very expansive. It enables the establishment of a cross-organization Grid with a hierarchical accounting architecture, in the manner of making use of existing accounting systems belonging to telecommunication industries, scientific and engineering computing communities, stocks and futures trade, and many more. A prototype of our framework in this paper has been implemented and will probably apply to the ShanghaiGrid project.

The further research will focus on how to implement more LAMs for existing accounting systems, giving a specification of the interfaces between GAM and LAM. Another issue will be to embrace our framework with OGSA.

References

1. I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
<http://www.globus.org/research/papers/anatomy.pdf>.
2. B. Rajkumar: Economic-based Distributed Resource Management and Scheduling for Grid Computing. A thesis submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, School of Computer Science and Software Engineering Monash University, Melbourne, Australia. April 12, 2002.
3. M. P. Rosario, G. Andrea, W. Albert: An accounting system for the DataGrid project - Preliminary proposal. www.globalgridforum.org/Meetings/GGF3/Papers/ggf3_11.pdf, September, 2001.
4. B. Alexander, B. Rajkumar: GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration, www.gridbus.org/papers/gridbank.pdf, 2003.
5. Z. Liang, L. Zhang, S. Dong, W. Wei: Charging and Accounting for Grid Computing System, 3th International Workshop on Grid and Cooperative Computing, 2003.
6. S. Newhouse: Grid Economic Services Draft, www.ggf.org, 2003.

Agent-Based Resource Selection for Grid Computing

Song Shen and Gregory M.P. O'Hare

University College Dublin, Belfield, Dublin 4, Ireland
{Song.Shen, Gregory.OHare}@ucd.ie

Abstract. Agent technology is critical in providing solutions to grid computing, including resource selection. Traditionally, agent deliberation offers a deductive process whose deliberation cost is very high and also difficult to measure. We consider the use of Multi-Agent Systems (MAS) for grid selection. Within this domain timing is an important index. Once the client applies for some service, the MAS should be able to quickly make a match. Furthermore, as autonomous software entities, MAS are expected to control its own deliberation behaviors. Within this paper, we propose agent-based resource selection with an agentbased fuzzy decision-making capability that enables better deliberation control and hence provides better selection solution.

1 Introduction

The recent explosion of interest in information sharing and Internet application has necessitated resource sharing and coordinated problem solving in a dynamic and multi-institutional environment. Grid computing is a new technology that has emerged at the end of last century that underpins distributed problem-solving solution.

Resource sharing, coordinated problem solving and dynamic multi-institution are basic characteristics of grid computing [8]. In a grid computing environment, resources can be computational resources, storage resources, network resources, and/or code repositories. These remotely distributed resources are integrated through communication with various kinds of security solutions.

A collective multiple-resource layer provides a variety of services including: directory, co-allocation, selection and scheduling, brokering, monitoring and diagnostics, data replication, grid-enabled programming systems, workload management systems and collaboration frameworks. Coordinating collective resources is a particularly complex high-level task whereby multiple resources are integrated into a wide-area distributed system [8]. Multi-Agent Systems constitute a highly suitable technology set for the effective provision of such services providing collaborative intelligence, autonomy, and social capabilities.

Manola and Thompson [2] were the first to propose the application of agent system in computational grids. They present different perspectives to grid environments and describe their system entitled Control of Agent-Based Systems (CoABS) grid. Functionally, the CoABS Grid knows not only about agents, but also about their computational requirements, and available computational resources. The CoABS Grid provides a unified but distributed computing environment within which computing re-

sources are linked seamlessly. The CoABS Grid also provides the infrastructure for large-scale integration of heterogeneous agent frameworks. Bradshaw et al. [3] have similarly focused upon the use of agents in order to simplify those problems inherent in grid computing. Rana and Walker [4] have demonstrated a good example of agent grid, while Foster et al. [5] [6] have presented an Open Grid Services Architecture that addressed the challenges in achieving various qualities of service issues when running applications on top of different native platforms.

It is anticipated that agent technology will help to provide reliable, scalable, survivable, evolvable, and adaptable systems. However, in order to provide solutions to grid computing, agent theory needs to solve several existing problems [1][9]. One of these is the controllability of the deliberation process. Traditionally, agent deliberation imitates the human cognitive process whose deliberation cost (usually means time cost) is very high and also difficult to measure. Weighing of different service providers is also difficult because of the existence of a myriad of selection criteria different criteria. In considering MAS for grid applications timing is an important index. Once the client applies for some service, the MAS should be able to quickly make a match. It may not care about the persistence problem, but there should be a time limit for selection. Therefore deliberation is considered as a time bounded deliberative process. Since there is no action within control loop, the control of deliberation process depends on the estimation and control of workload within deliberation and perception.

This paper primarily concerns about the MAS decision-making on resource selection for grid computing. We will start with related work, follow by Agent Fuzzy Decision-Making (AFDM) [1] and its controllable solution to decision-making, and end with a summary and future work.

2 Related Work

Rao & Georgeff [10], Woodriddle [9] and Singh [15] have all developed logic theories involving multiple worlds and logical formalisms. In their definitions, each world is viewed as a combination of time and state expressions. The logic theories greatly strengthen the usability of Belief Desire Intention (BDI) and are recognized as critical compositions in the BDI family. However, it is still difficult to integrating the current logical family with practice.

Apart from the mainstream logical formalization, applying CBR (Case-Based Reasoning) onto multi-agent system is a strong experience-based approach on BDI practice in recent years [11][12]. The reasoning is based on the reuse of past experiences or cases. Cases in CBR are represented by a triple of 'problem', 'solution of the problem', and 'outcome'. 'Outcome' is the resulting state of the world when the solution is carried out and will be reused as a basis for future problems that present a certain similarity, as the basic principle of CBR defines.

QDT (Qualitative Decision Theory)[13] [14] [16], on the other hand, provides another decision solution different from traditional BDI logic. QDT is a multi-level qualitative approach developed to reason about uncertainties, which are typically represented by a plausibility function. QDT is theoretically complete. The key to QDT

application is, however, how to remove the uncertainties or to calculate the possibilities in a broad sense, and in what way we can integrate experience with BDI model.

The agent deliberation process can be viewed as an inherently fuzzy decision-making process. By using the word *fuzzy* we mean when we try to think out a solution, we usually select several goals and decide by weighing them on certain aspects we care about, such as cost, time, quality and accessibility etc, together with our preference. E.g., while we want to select a place to travel from a group of candidates, we think about cost, the time needed, how easy it is to get there (transportation), and expected enjoyment (quality of the result) etc; while we prepare our career to be an academic, we need to think about how much money to pay, how many years to spend on studying, how easy to become an academic, and expected career when successfully becoming an academic etc. On most occasions, it is more effective to weigh corresponding aspects in a fuzzy way, if the experience values are at hand.

AFDM is fuzzy-logical based deliberation model. AFDM addresses the limitations of present formalisms within BDI models by making decisions based on quantified fuzzy judgment. The AFDM matrix model enables quantitative calculation and thus provides a more practical solution to BDI models. In addition, more flexible and controllable solutions to BDI persistence and incremental decision-making can be expected with the introduction of AFDM.

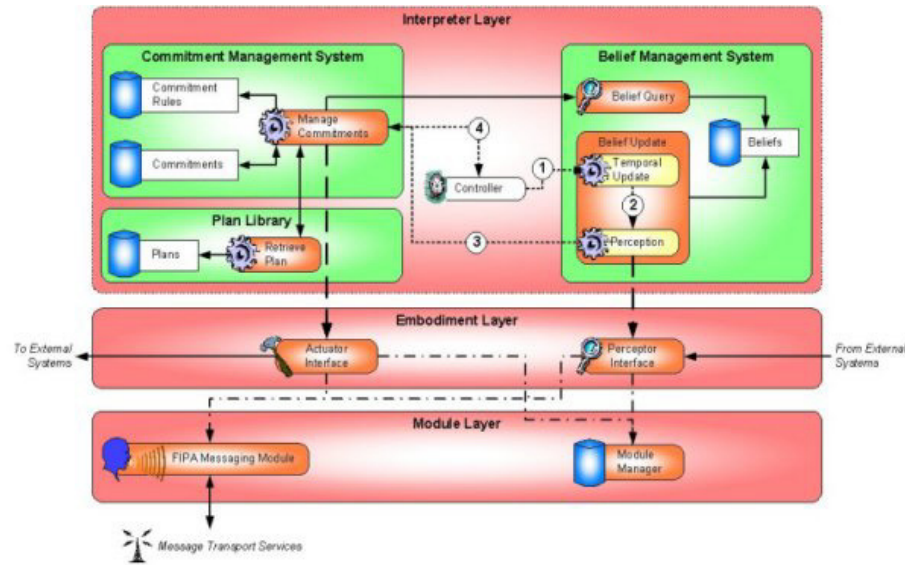


Fig. 1. Agent Factory interpreter layer

Agent Factory is a cohesive framework that delivers structured support for the development of agent application and deployment of agent-oriented applications. Specifically, it is realized over four tiers, the Agent Factory Agent Programming Lan-

guage(AFAPL), the Runtime Environment(RTE), the Development Environment(DE), and the Development Methodology(DM). AFDM is a sub platform on Agent Factory.

The key components of this architecture are:

- The Commitment Management System (CMS).
- The Belief Management System (BMS)
- The Plan Library (PL)
- The Controller
- The Actuator Interface
- The Perceptor Interface
- The Module Manager
- The FIPA Message Queue

As one of the MAS' most critical tasks, resource selection takes the duties of searching available servers on the Internet, weighing them according to certain criteria, and selecting the most optimum server or server groups to take the task. Usually, resource selection needs to work with constraints of user and service-provider within limited period of time. It means selection is required to work under a controllable way. Since agents are autonomous entities embedded in the environment, the ability on controlling their own deliberation process is one of the most critical indexes to measure agent behavior.

The traditional agent BDI interpreters have difficulties in dealing with time-limited deliberation since their deliberation costs are unknown and their control loops are uncontrollable. Cost estimation is essential to the control of deliberation. AFDM achieves control of deliberation by introducing a matrix model and separating each task into controllable part and uncontrollable parts and embedding only the controllable part into the control loop.

Our primitive desire is to build up a resource selection mechanism based on a kind of agent fuzzy thinking mode (by deliberating on multiple aspects of goals). Such a model will, to some extent, address the limitations of the present BDI model to resource selection applications of grid computing. This flexible platform can integrate with other experience-based techniques within an AFDM interpreter.

The framework includes multiple resources or parallel applications. However, in order to simplify the theoretical explanation, the deployment of MAS is explained as the functioning of AFDM on a single resource at this time. We omit the part of parallel computing in our following text.

3 MAS Decision-Making with AFDM

3.1 Definitions

The BDI model serves as a first-order prototype that leaves much to be further developed. In a typical BDI agent architecture, the agent states of are represented with 3 component types: Desires, Beliefs and Intentions. Here are some basic definitions that we commission within our model. Intentions are viewed as those goals that an agent has committed to achieve. Belief is the information an agent has about the world. The

information could be about its internal state or about the environment. In terms of our model, the belief is mainly from the library that contains server information, which is dynamically and periodically updated by agents. This information includes varying different aspects of servers relevant to selection, communication ability, calculation ability, quality of service and economic cost. Desires denote states that an agent wishes to bring. Specifically within this paper, desire is a multi-axis reference frame in which the axes are independent to each other and each axis represents an aspect typically associated with decision-making. The number of axis corresponds to the decomposition of desire. Goals are the desire-consistent beliefs projected in desire space. They are mapped onto the desire space and measured on these corresponding aspects. If a goal has successfully passed through a weighing function and is chosen by an agent as an intention, we say that the agent has made a commitment to this goal.

Expressing goals with a set of aspects borrows from models of human cognition. We usually weigh different goals by comparing different aspects that we care about most. For certain kinds of decision-making, those aspects with which we are concerned can be drawn from several indexes that human beings commonly care about most. The world of desire is serial, Euclidean, and also dynamic. A typical desire space can be denoted via $\vec{D} = (\vec{D}_1 \ \vec{D}_2 \ \vec{D}_3 \dots \vec{D}_N)^T$, in which the desire vector bases $\vec{D}_1 \ \vec{D}_2 \ \vec{D}_3 \dots \vec{D}_N$ stand for aspects of desire. The decomposition of desires follows human cognition and quantification necessity. These aspects can be further decomposed for weighing necessities. The I-th goal G_I is denoted as $(G_{I,1} \times \vec{D}_1, G_{I,2} \times \vec{D}_2, \dots, G_{I,N} \times \vec{D}_N)^T$, where $G_{I,1}, G_{I,2}, \dots, G_{I,N}$ each represents the corresponding scores (or rankings) on concerned aspects of I-th goal. For example, $G_{I,j}$ stands for the score (or ranking) of I-th goal on J-th aspect

3.2 AFDM Interpreter

Practical reasoning consists of two major activities. The first is deliberation, deciding what to do; the second is planning or means-end reasoning, deciding how to achieve the intention. The relationship of agent deliberation, Beliefs, Desires and Intention are depicted in Figure 2.

Generally speaking, there are limitless desires in an agent system at the same time, some strong, some weak. Although desires are possibly inconsistent, the goals generated from beliefs and desires are required to be consistent, and achievable within our approach. More specifically in this paper, we pay particular attention to the decision-making of a group of consistent goals.

Planning is also an important component of an interpreter. Traditionally agent engineers tend to associate plans with intentions. That is, an intention will be further planned into possible actions after an intention is generated. However, theoretically planning and deliberation are often viewed as inextricably linked. Separation of the two processes significantly lessens the working burden but causes theoretical problems. Alternatively, deliberation cannot ensure best planning result. In this paper, the whole reasoning process is simplified by associating plans with goals that are selected

after fuzzy deliberation, although we sometimes still explain deliberation and planning separately for conceptual considerations.

We adopt formalisms similar to those of Wooldridge [9] and define Des, Bel, Int, Act, Gal correspondingly as all possible desires, beliefs, intentions, actions and goals respectively. B, D, I are the states of a BDI agent at any given moment constituting an agent state triple $\langle B, D, I \rangle$, where $B \subseteq \text{Bel}$, $D \subseteq \text{Des}$, and $I \subseteq \text{Int}$, while goal G is a mixing state, $G \subseteq \text{Gal}$. g denotes an arbitrary element of goal matrix, $g = G_{I,J} \mid I \in \{1, N\}, J \in \{1, M\}$. In addition, we denote S as an arbitrary set, $\wp(S)$ is the powerset of S.

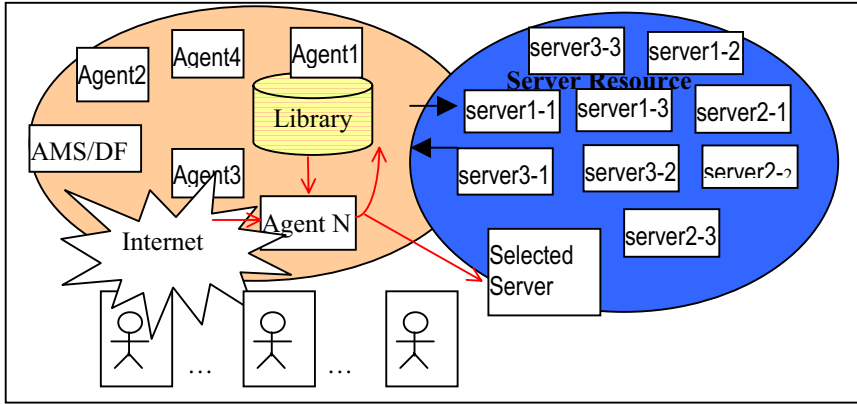


Fig. 2. Grid computing architecture

The general interpretation procedure of AFDM can be described as follows. At the beginning of deliberation, the option generating function (function Opt) reads the user service application and perceives the environment (available server set) to get beliefs, retrieving a list of possible goals (application-appropriate subsets) and corresponding information for further deliberation; then the goals are mapped onto desire space and assigned with real values from experience and beliefs by a mapping function (function Map); the goals are further filtered through constraints provided by user and service-provider by an embedded filter function (function Flt) to generate grid resource group; and the best goal or goal group is selected through synthesis weighing all the surviving goals together with a plan of actions and is committed as intention (function Wgh). The agent will assign the task to the selected server (decided by intention) later on.

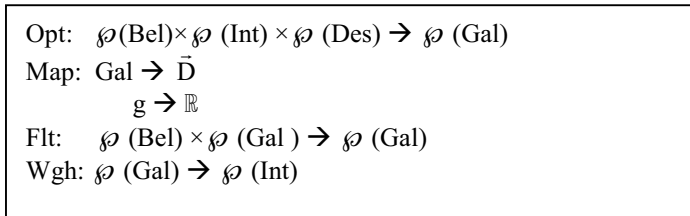


Fig. 3. AFDM functions for resource selection

Interpretation can be modeled with main functions listed in Figure 3.

A unique characteristic of our model is the mapping of a set of goals onto a desire space and the subsequent assignment of each element of the goals with a real value, enabling the quantitative weighing of different goals.

3.3 Fuzzy Matrix Model

The main task of the deliberation process is to weigh different goals. In the desire space proposed in this paper, goals are weighed by their Euclidean lengths, which are comprehensive magnitudes of the goals on multiple desire bases. We use goal vector $\overrightarrow{G^D(M)}$ to express mapping of goals in desire space, quantitative vector $\overrightarrow{Q^D(M)}$ to denote the mapping of goal group in desire space with preference. Each goal group $G(M)$ is associated with a goal vector and a quantitative vector.

$$\overrightarrow{G^D(M)} = G(M, N) \times \overrightarrow{D(N)} \quad (1)$$

$$\overrightarrow{Q^D(M)} = W^G(M, M) \times G(M, N) \times W^D(N, N) \times \overrightarrow{D(N)} \quad (2)$$

Goals are represented as vectors originating from the origin in the desire reference frame. The fuzzy commitment rule is based on the measurement of the magnitudes of goal vectors in desire space, $|\overrightarrow{Q^D(I)}|$, where $I \in (1, M)$. Even though AFDM is a fuzzy process, the vectors and matrixes are not necessarily be unitary ($[0, 1]$ as defined by fuzzy logic) since we only weigh goals for comparison or weighting, rather than measure their actual values. Mathematically, we can use Euclidean length to measure the magnitudes of vectors, which could be time-consuming with a big group of options. So alternatively, on most occasions, we use Manhattan distance to measure the goals.

$$|\overrightarrow{Q^D(I)}| = \sum_{J=1}^N W^G_{I,I} \times G_{I,J} \times W^D_{J,J} \quad (3)$$

Where $W^G_{I,I}$ and $W^D_{J,J}$ are $I \times I$ and $J \times J$ weight matrixes on goals and on desire aspects. Weight matrixes can be used to adjust the importance of different application genres since we adopt a uniform weighting criteria. Suppose L is the 1st empty element in the intention queue and I_L is the L -th intention, then the top ranking goal G_J is committed as I_L . Formally,

$$I_L = \{ G(M) \mid \arg \max(|\overrightarrow{Q^D(J)}|) \wedge J \in \{1, M\} \} \quad (4)$$

The service application is subsequently launched upon the selected (intention) grid resources.

4 Controllable Resource Selection

4.1 Aspects Settings

We now provide the main selection parameters to be considered in server selection:

- Cost ($W_{1,1}^D$): Denotes how much the client needs to pay for the service. It is usually in the form of a price.

- Calculation ability ($W_{2,2}^D$): Denotes the ability of service-provider to this specific service, depending on its mathematical ability, number of CPUs available (including the present workload), operating system, processor type and speed, available memory, and storage etc.

- Communication ability ($W_{3,3}^D$): Expressed in the form of communication bandwidth and latency. If the service cares about wireless servers, then the communication ability should also include communication quality.

These three basic parameters can if necessary, be decomposed further. For example, calculation ability can be decomposed as mathematical ability, number of CPUs available, operating system, processor type and speed, available memory and so forth. In addition the MAS also undertakes the duty of distributed monitoring to order to track and forecast dynamic resource conditions. We can also add a new aspect,

- Quality of service ($W_{4,4}^D$): Denotes the overall evaluation of service quality.

Among those aspects considered above, some are less dynamic than others, e.g. cost, mathematical ability, quality of service, while some others are highly dynamic, for example, calculation ability and communication ability, in detail, available CPUs, memory, storage and communication bandwidth with which data can be sent to a remote host.

4.2 Cost Estimation

Some designers may worry about the increase of workload in processing matrix data. Not surprisingly, precision and workload are always the contradictory forces. It may become a critical problem when N , and M increase. The calculation necessary for solving the matrix will be approximately proportional to $N \times M$. The workload can be calculated once the dimensions of the matrix are determined.

Except for weighting aspects or goals. The weight matrix is sometimes also adopted to mask options or aspects. In the main, we use an incremental matrix solution to adjust the dimension of matrixes instead of expending lots of efforts to change matrix volume in the real environment. That is, empty coefficients will be applied to certain goals or aspects when they are no longer useful in processing. The interpreter will skip the calculation on goals or aspects if the corresponding coefficients are 0. Whenever a new option or new aspect emerges, the interpreter will first search an option library to check if there are empty rows or columns. If yes, then the new option or aspect will be filled into the vacancy. This ensures that a dynamic customization of the decision making process can be achieved either introducing more or less precision in order to yield a more or less system responsiveness.

The ability to time estimation deliberation and planning cost is critical in order that an agent may manage the timing of the process and thus may be to wholly in control of the deliberation process. Within traditional agent deliberation and planning it is hard to solve this problem because of unknown workload. In contrast, control is achievable with the adoption of AFDM. Let's see how AFDM solve the problem within a MAS domain.

4.3 Multi-agent Cooperation and Time-Limited Deliberation

Let us first analyze the contents contained the basic reasoning depicted in Figure 4, where T and T_0 are time cost and time limit of resource selection, and $Est()$ is a cost estimation function. The workload primarily consists of two processes: perception of environment and the decision-making process. In our MAS, perception and decision-making (reasoning) are distributed among different agents. The agent working on decision-making only weighs to get the best solution for the specific service, leaving the perception for other agents. Actually every agent, in her spare time, concentrates on the task of updating server information (perception) into the library. The principle of agent cooperation is shown in Figure 2. Thus within AFDM, only the decision-making process is a time-cost process in the control loop, while the dominant cost in the decision-making process is the weighing of server options on certain aspects. Moreover, the reasoning process deals with the calculation of matrixes that are fixed once the numbers of aspects and goals are determined. This is a peculiar merit of AFDM interpretation.

```
//Practical reasoning for grid computing
B, I, D, G, Wg, Wd, T ← B0, I0, D0, G0, Wg0, Wd0, T0;
While not (Empty() or Succeeded() or Impossible()) do
  B ← Brf(B, p);
  G ← Opt(D, B, I);
  T0 ← Est(N, M);
  While (not Succeed() and T < T0) do
    G ← Map(D, G);
    G ← Flt(B, G);
    I, P ← Wgh(I, G);
  End while
End while
```

Fig. 4. Practical reasoning for resource selection

Decision-making results and weighing result will be again stored in the option library in the form of cases. This provides a mechanism by which basic learning may improve the intelligence of the MAS through each decision. Besides time-limited deliberation, adjacent applications can refer to previous weighing results. We refer to this as incremental decision-making (ID). Since some aspects change slowly, ID can

adopt either the weighing results on some reluctant aspects or only a certain number of top options that have survived from previous selection. Practical solutions can be more flexible.

Through the adoption of AFDM, time-limited deliberation can be achieved at the estimation of the workload. The deliberation cost is approximately proportional to $N \times M$. By estimating the deliberation cost, we can adapt to the time constraints by adjusting number of options or number of aspects involved within the deliberative process.

4.4 Example Scenarios

The following simplified example will be adopted to explain how AFDM works on resource selection of grid computing.

Table 1. Resource selection example

Ranking Scores Options	Communication Score (rank)	Calculation Score (rank)	Cost Score (rank)
A	5.0(1)	4.0 (2)	1.0 (5)
B	2.0(4)	2.0 (4)	3.0 (3)
C	3.0(3)	5.0 (1)	2.0 (4)
D	4.0 (2)	3.0 (3)	4.0 (2)
E	1.0 (5)	1.0 (5)	5.0 (1)
$\mathbf{W}_{J,J}^D$ $J=1,2,3$	25%	35%	40%

Suppose there are 5 available servers (groups) on the Internet. Resources are weighed on aspects of cost, calculation ability and communication ability (we ignore the ‘Quality of service’ aspect purely for simplification). For a practical application, these aspects are further decomposed for more detailed comparison. Here we only use these 3 aspects in order to provide a simple animation of the AFDM approach.

Let’s detail the characteristics of the 5 choices, see Table 1. Among them, A is the easiest to communicate with (so rank 1 and score 5 on this aspect) and of high calculation ability (score 4), but the most expensive (score 1); E is the cheapest option (score 5), but not good in its communication ability and calculation ability (so rank 5 and score 1 on both); while D, although not the top selection on any aspect, is the only candidate that is above the average level on all aspects; B and C are medium choices on each aspect. The deliberation process empowers the agent enabling it to decide which choice to select by weighing the 5 possible choices with corresponding coefficients and the scores (or ranking) of each choice on each evaluation aspect. $\mathbf{W}^D(N, N)$ is first decided upon by empirical data and is dynamically improved with the increase of application cases. Scores (ranks) are adopted to weigh the options because they are much easier to decide than quantitative values.

By applying fuzzy decision-making within the example shown in Table 1, we get:

$$\overrightarrow{Q^D(M)} = W^G(M, M) \times G(M, N) \times W^D(N, N) \times \overrightarrow{D(N)}$$

$$= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} 5 & 4 & 1 \\ 2 & 2 & 3 \\ 3 & 5 & 2 \\ 4 & 3 & 4 \\ 1 & 1 & 5 \end{bmatrix} \begin{bmatrix} 25\% \\ 35\% \\ 40\% \end{bmatrix} \begin{bmatrix} \overrightarrow{D_1} \\ \overrightarrow{D_2} \\ \overrightarrow{D_3} \end{bmatrix}$$

Where $\vec{D} = (\vec{D}_1 \vec{D}_2 \vec{D}_3)^T$ correspond to bases of communication ability, calculation ability and cost respectively. The vector magnitudes of the five goals are calculated into the following matrix,

$$|\overrightarrow{Q^D(M)}| = (3.05, 2.40, 3.30, 3.65, 2.60)^T$$

The reasoning results are then ordered from (A, B, C, D, E) into (D, C, A, E, B). D gets the maximum score and is then selected by fuzzy reasoning as an intention. The intention is associated to the corresponding plan with plan descriptors. So intention will then trigger a selection of matching the task to server D.

5 Summary and Future Work

Within this paper we have proposed an AFDM based resource selection for grid computing applications. In our model, *desire* is a multi-axis reference frame in which each axis represents an aspect of human wish, and goals are weighed by mapping these onto concerned desire bases. Within this model, solutions at divergent quantitative levels are achievable. The solution offered by AFDM obviates the limitations of present BDI models by introducing a fuzzy matrix decision-making model. More specifically AFDM, as a fuzzy decision-making matrix model, can serve at a cost-controllable mode. Hence we are quite confident of its usefulness in grid computing applications.

Our approach is merely a preliminary attempt at adopting a controllable BDI interpretation process to resource selection within grid computing. Although the practical application of AFDM framework to grid computing is still on-going, we have already formulated a detailed solution and a series of simulations have proven very promising. Implementation details together with a series of experiments conducted within the Agent Factory platform, will be discussed in our later works.

Acknowledgements

Gregory O'Hare and Song Shen gratefully acknowledge the support of Science Foundation Ireland under Grant No. 03/IN.3/1361.

References

1. Shen, S., O'Hare, G.M.P., Collier, R.: Decision-Making of BDI Agents, a Fuzzy Approach, to be published on The Proceedings of The 4th International Conference on Computer and Information Technology (CIT2004), Wuhan, China, Sept., 2004
2. Manola, F., and Thompson, C.: "Characterizing the agent grid", <http://www.objs.com/agility/techreports/990623-characterizing-the-agent-grid.html>, June 1999.
3. Bradshaw, J. M., Suri, N., A. Cañas, David J. R., Ford, K., Hoffman, R., Jeffers, R., and Reichherzer, T.: Terraforming cyberspace, *Computer*, 34(7):48–56, July 2001.
4. O. F. Rana and D. W. Walker: The Agent Grid: Agent-based resource integration in PSEs, In Proceedings of the 16th IMACS World Congress on Scientific Computing, Applied Mathematics and Simulation, Lausanne, Switzerland, August 2000.
5. Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S.: The physiology of the Grid: An open Grid services architecture for distributed systems integration, <http://www.globus.org/research/papers/-ogsa.pdf>, January 2002.
6. Foster, I., Kesselman, C., and Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organizations, *International Journal on High Performance Computing Applications*, 15 (3):200–222, Fall 2001.
7. Wooldridge, M., and Jennings, N. R.: Intelligent agents: Theory and practice, *The Knowledge Engineering Review*, 10(2): 115–152, 1995.
8. Overinder, B.J., Wijngaards, N.J.E., Steen, M.van, and Brazier, F.M.T.: Multi-Agent Support for Internet-Scale Grid Management, <http://www.cs.vu.nl/pub/papers/globe/aisb-grid.02.pdf>.
9. Wooldridge, M.: An Introduction to Multiagent Systems, John Wiley & Sons (Chichester, England). ISBN 0 47149691X, February, 2002
10. Rao, A.S. and Georgeff, M.P.: BDI Agents: From Theory to Practice, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, 1995
11. Corchado, J. M. Laza, R.: Construction of BDI Agents from CBR systems, 1st German Workshop on Experience Management, Lecture Notes in Informatics, Berlin, 2002
12. Olivia C., Chang C. F., Enguix C.F. and Ghose A.K.: Case-Based BDI Agents: An Effective Approach for Intelligent Search on the World Wide Web, AAAI Spring Symposium on Intelligent Agents, Stanford University, USA, 22-24 March 1999
13. Boutilier, C.: Towards a logic for qualitative decision theory, In Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning (KR'94), 75–86, Morgan Kaufmann, 1994
14. Pearl, J.: From conditional ought to qualitative decision theory, In Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI'93), 12–20, John Wiley and Sons, 1993
15. Singh, M. P. and Asher, N. M.: Towards a formal theory of intentions, In Logics in AI — Proceedings of the European Workshop JELIA-90 (LNAI Volume 478), pages 472–486. Springer-Verlag: Heidelberg, Germany, 1991
16. Dastani, M., Hulstijn, J. and van der Torre, L.: BDI and QDT: a comparison based on classical decision theory, In Proceedings of AAAI Spring Symposium GTDT'01, 2000.

Visualization Practice in Construction of a Computational Grid for Multidisciplinary Design Optimization of Aero-crafts*

Hong Liu¹, Xin-hua Lin², Xi-li Sun², Xiao-Feng Qi², Qian-ni Deng², and Xin-da Lu²

¹ Department of Engineering Mechanics, Shanghai Jiao Tong University

² Department of Computer Science, Shanghai Jiao Tong University

200030, Shanghai, P.R. China

hongliu@sjtu.edu.cn

Abstract. To construct a user orient computational grid for multidisciplinary design optimization (MDO) of aero-crafts based on web service, visualization of the computing data and visual steering of the whole process for MDO is highly required. In this paper, our visualization practice in construction of the Grid for MDO of aero-crafts is described. There are two kind of practice will be described in detail, including visual steering scheme for MDO and Visualization service for computing data. The visual steering scheme is constructed under web service environment and is performed as a series of web pages. Visualization Toolkit (VTK) and Java are adopted in visualization service to process the results of MDO of geometry and the CFD data. All visualization practice in construction of this grid makes the process for MDO of aero-crafts efficiently.

1 Introduction

Multidisciplinary Design Optimization (MDO) is highly required in the development of novel high-performance aero-crafts, including flight performance, aerodynamic performance etc. Therefore, for many years MDO of aero-crafts has been a significant challenge for scientists and designers.

Generally, MDO of a complex system, especially an aero-craft system, usually involves various disciplines. As the requirement for a complex system is increased, MDO problems become more complex. Advances in disciplinary analysis in recent years have made those problems worse and those analysis model restricted to simple problems with very approximate approaches can not been used again. As those analysis codes for MDO of flight vehicles have grown larger and larger, it is indeed too incomprehensible and difficult for a designer-in-chief to maintain. Therefore, the role of disciplinary scientist increases and it becomes more difficult for a designer-in-chief to manage the design process. To complete the design process smoothly, the designer-in-chief must joint all specialists in a collaborative optimization process. Thus,

* This research is supported by the National Natural Science Foundation of China (Grant number: 90205006, 60173013) and Shanghai Rising Star Program (Grant number: O2QG14031).

a need exists, not simply to increase the speed of those complex analyses, but rather to simultaneously improve optimization performance and reduce complexity [1].

As the Grid technology developed, there appears a new way to solve the problem. In Grid computing environment, the computing resources are provided as Grid service. Then, we can construct a Computational Grid for MDO, to improve the performance of the MDO algorithms and gain many new characteristics that are impossible in traditional means for MDO of a large aero-craft system. This computational grid for MDO can make the design process be easily controlled and monitored and can be conducted interactively, and through this Grid system many specialists in various disciplines can be easily managed in one MDO group.

To construct a user orient computational grid for MDO of aero-crafts based on web service, visualization of the computing data and visual steering of the whole process for MDO is highly required. Presently there exists much software for distributed MVE (modular visualization environments), such as AVS/Express (Advanced Visual System), IBM's OpenDX (free source) and IRIS Explorer. As for the function, all those MVE toolkits can be adopted in this paper. However, the computational grid should have the characteristic for trans-platform. Then Java is the right developing toolkit for our grid system. To construct this Java system, it seems that Visualization Toolkit (VTK) is more attractive choice, which can be built as two layer architecture: one is kernel layer for pre-compiling and the other is packaging layer with interpreting computer language.

Therefore, in this paper, our visualization practice in construction of the Grid for MDO of aero-crafts is described. There are two kind of practice will be described in detail, including visual steering scheme for MDO and Visualization service for computing data. The visual steering scheme is constructed under web service environment and is performed as a series of web pages. Visualization Toolkit (VTK) and Java are adopted in visualization service to process the results of MDO of Geometry and the CFD data. All visualization practice in construction of this grid makes the process for MDO of aero-crafts efficiently.

In the following section, the framework of the computational grid for MDO will be described briefly. Then, the visual steering scheme for MDO and Visualization service for computing data will be discussed in details in the third section and in the forth section respectively.

2 Computational Grid for MDO of Aero-crafts

2.1 Computational Grid Framework for MDO of Aero-crafts

Procedure for Multidisciplinary Design Optimization (MDO) of aero-crafts by using genetic algorithms (GA) can be completed with the following steps. Firstly, set the range of a set of given designing parameters and use a random algorithm to select a specific value for each parameter from the given range. Secondly, use the analysis codes to compute each child task and get the result set. Thirdly, use a comparison algorithm to select the best result and use the best as the "seeds" to generate the next generation.

The Computational Grid for MDO is composed of many Grid services. Each Grid service is built on the web and grid technology. A Grid service is a web service that conforms to a set of conventions (interfaces and behaviors) that define how a client interacts with a Grid service [2]. Each Grid service has the responsibility to maintain the local policy.

2.2 Components of the Computational Grid for MDO

The Computational Grid System for MDO is composed of three modules as shown in figure 1. The first part is User service module. As shown in the figure 1, the executor and the co-designer are the specific users of this system. The user can submit the tasks, monitor the computing progress and get the middle and final results via the web pages. When received the requests from the users, the User service can query the UDDI center to find what kinds of services available now. The User service can supports the user accounts management, supports the task submitting and parameter adjusting. It can interact with other services to finish the computing work. The second part is the UDDI center module. The responsibility of this module is acting as an information exchange center. The third part is the Application services, including Allocate service, Select service and Analysis service including CFD service, CFM service and other Analysis services.

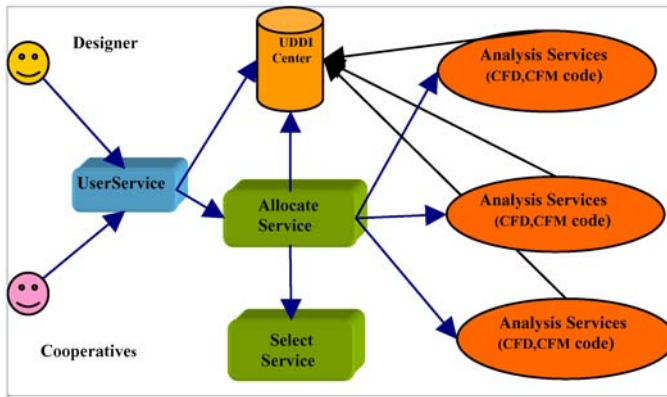


Fig. 1. Computational Grid System Framework.

3 Visual Steering Scheme in the Computational Grid for MDO

The computational grid for MDO finally works well by connecting users and the Grid services. For users, if they want to start a new design, they only need submit their tasks through the web. To complete a MDO task, the designer-in-chief also can communicate with other designers from various disciplines through the web. There is a billboard on the web page. The task executor and the co-designers exchange information by it. When the co-designer finds some parameters are not suitable for the task,

he leaves messages on the billboard. The task executor can adjust the parameters. Web services will help any designers complete their whole designing tasks mutually and interactively.

To provide users conduct their jobs from the special web server of the computational grid, a User Service with visual steering is constructed. The User service is has following functions: (1) Support the user accounts management; (2) Support the Task submitting, parameter adjusting, process monitoring and the results obtaining; (3) It can interact with others services to finish the computing work; (4) Provide on-line instruction for users.

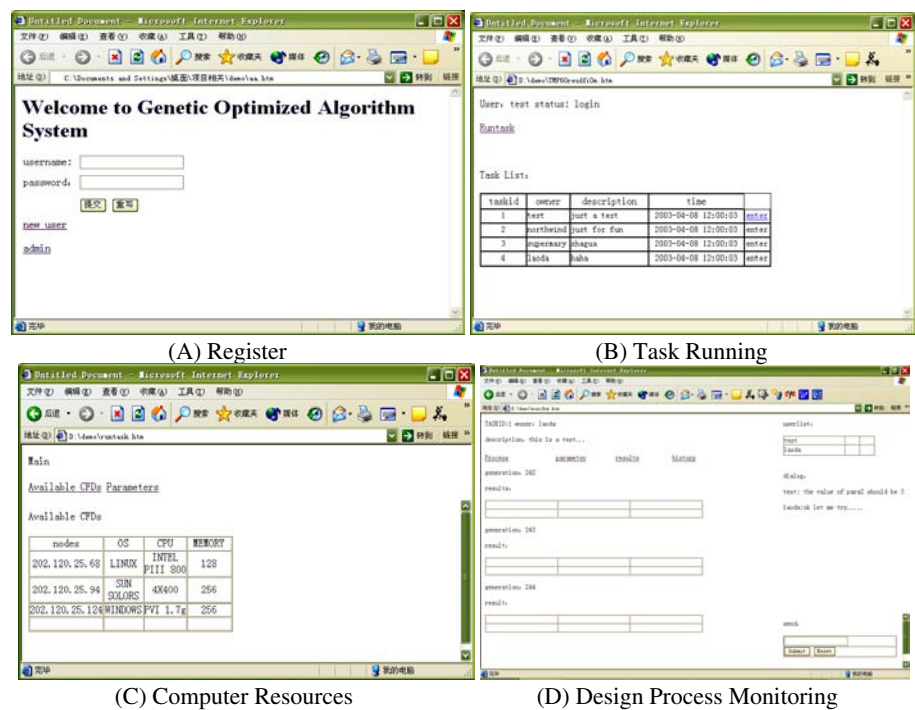


Fig. 2. The prototype web pages of the Web services of the optimization Grid system.

When a user login in the system via a Java Server Page (JSP), the daemon process of the web server redirect the request to a account manage process. When an authenticated user submits a task, the tasks control process will allocate a sequence number for that task and generate a record in the database. All the information of the task including the parameters, the results and the interaction messages between the executor and co-designers can be record in the database as needed.

There is a billboard on the web page, with which the task executor and the co-designers can exchange information. When any co-designer finds some parameters are not suitable for the task, he can leave messages on the billboard. The task executor can adjust the parameters by the JSP that set parameters after the co-designer's points of view are carefully considered.

Figure 2. shows some prototype web pages for the computational grid for MDO. (A) illustrates the first web page entering the Grid system, which is a register service and is also an accounts manager service. (B) illustrates the second web page of the Grid system, which gives the choice and service to run an optimization design task. (C) is the third web page for the Grid system, in which computer resources checking can be conducted. In the (D), the service to monitor the design process is furnished in the Grid system. From Figure 2.(A) to Figure 3.(D), it is clearly that what a designer to do is obtain all his requirements from a special Web.

4 Visualization Service in the Computational Grid for MDO

Post-processing of MDO and Computational Fluid Dynamics (CFD) data is a very important aspect of scientific visualization. In this section, we describe how the Visualization Service be constructed in the computational grid for MDO of aero-crafts. It integrates the advantages of the distribute visualization and grid service technology.

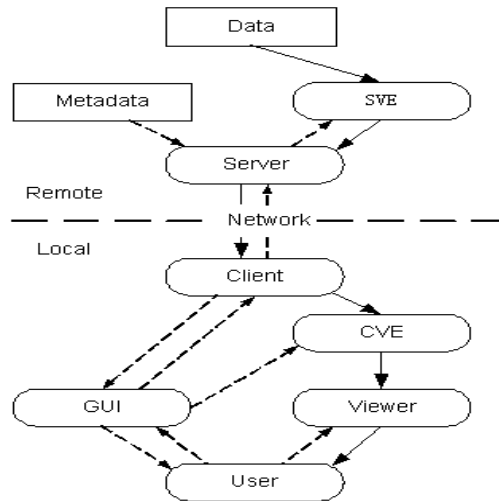


Fig. 3. Distributed Visualization System Architecture.

As shown from figure 3., in this distributed visualization system of grid, computing data and metadata are saved in the server section and user, GUI interface and viewer are saved in the client section. visualization engine can be divided into two parts, one part locates in the server section, named as SVE (*server-side visualization engine*), and another part locates in the client section, named as CVE (*client-side visualization engine*). If SVE transports data to the server, visualization process will be completed by the client section; or SVE complete the visualization process of the data, and CVE transports data and 2-D image to be viewed through viewer.

According to the basic instruction of distributed visualization system and the characteristic of the Grid for MDO, Visualization Toolkit (VTK) and Java are adopted

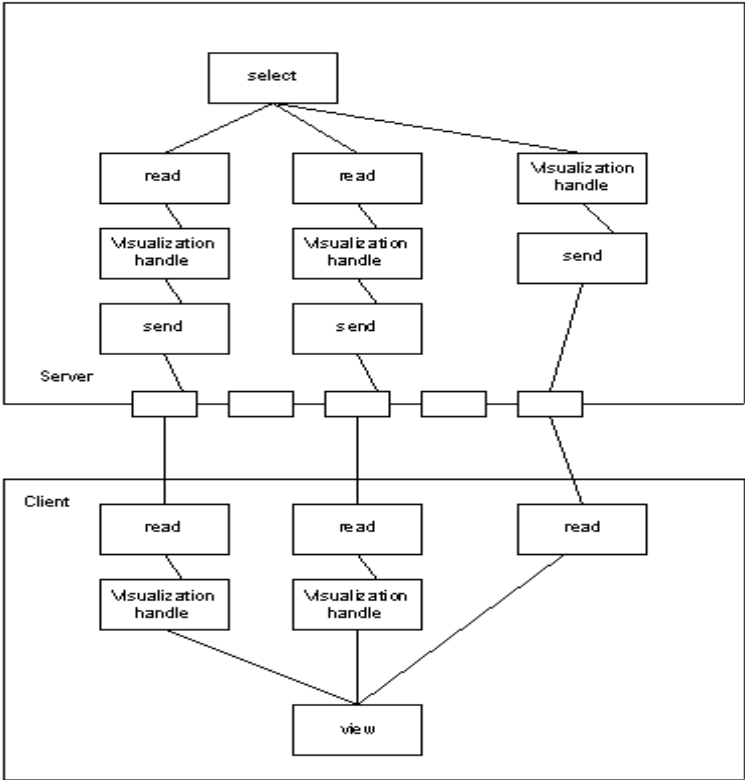


Fig. 4. Visualization Service System Architecture of the Computational Grid for MDO.

into the Visualization Service to process the optimized and CFD computation data. Figure 4 is the demonstration for Visualization Service System Architecture of the Computational Grid for MDO in this paper. The interface of the grid service is described by WSDL, and the internal implementation of the grid service is designed by using VTK as the visualization kernel. Middleware system used to design the grid computing environment is GT3, which is based on OGSA.

In this Visualization Service System, there are those components as following. Firstly, Preliminary Data Module, including data selecting module (SelectFile), data preprocessing module (ReadFile) and data sending module (SendData). Secondly, Visualization Module, including Iso-surface processing module (Isosurface), mesh processing module (Mesh), streamline processing module (Streamline), contour processing module (Contour), vector data processing module (Vector), boundary processing module (Boundary), slice and clip processing module (Slice & Clip) and slice shrink processing module (Shrink).

As can be shown from figure 5, it is the streamline of flow field around an aircraft with visualization system in this paper. As for the algorithms in the Visualization module, the Marching Cubes (MC) method is used to compute the iso-surface and the particle track method is used to compute the streamlines.

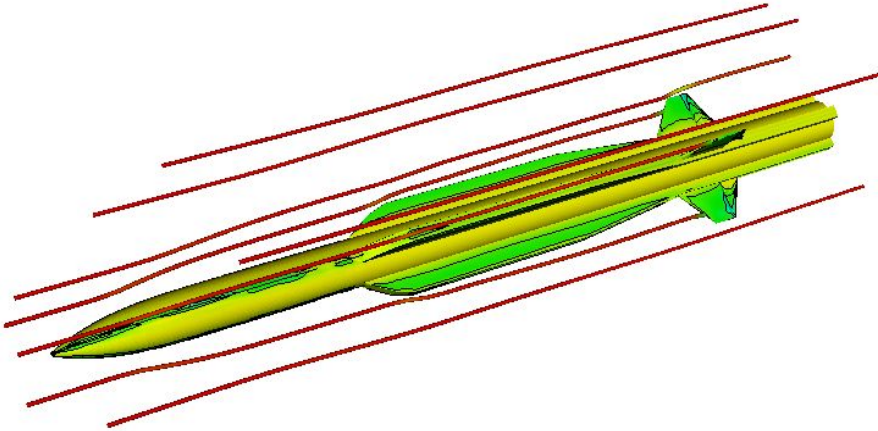


Fig. 5. Streamline of Flow Field Around an Aero-craft with Visualization Service System.

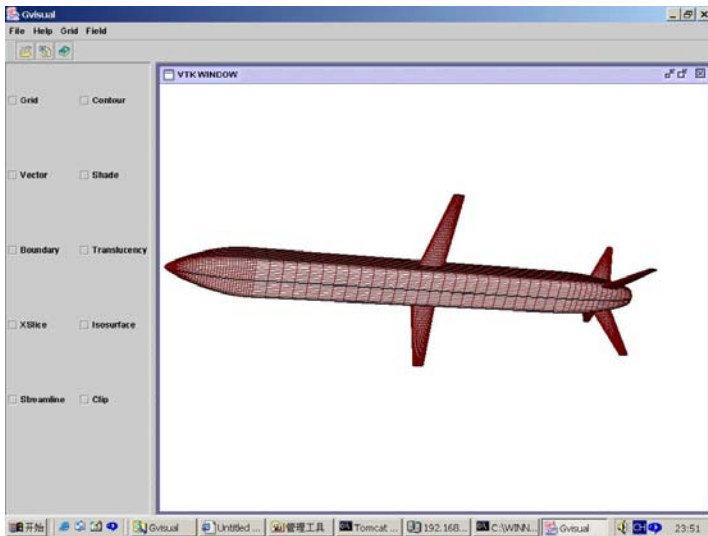


Fig. 6. The GUI in the client section of this Visualization System.

Figure 6 is the GUI in the client section, user of the computational grid for MDO can download it from the result monitoring web page. The experimental result indicates that the Visualization Service can use remote powerful computing resources to provide strong visualization transactions for clients. And it is also shown that all visualization services by our VTK based software can make the process for MDO of aero-crafts efficiently.

5 Conclusion

Visualization of the computing data and visual steering of the whole process for MDO is very important to construct a user orient computational grid for MDO of aero-crafts based on web service. Through implementation the visual steering scheme for MDO and Visualization service for computing data, we realize resource sharing and problem solving in dynamic, multi-designer and interactive way in the computational grid. All visualization practice in construction of this grid makes the process for MDO of aero-crafts efficiently. The advances of this computational grid for MDO can be clearly shown. Firstly, it presents a novel framework for the applications of MDO of aero-crafts, which can utilize the computing power over a wide range, and in which analysis service and visualization service served as web services can be dynamically changed. Secondly, it is a great improvement in the optimization design method for the aero-craft shapes. By using this Computational Grid system, designers can control the parameters during the progress and can also check the middle results. The designer and the co-designers can exchange their points of view about the designing. This enables scientists from various disciplines to complete collaborative design work interactively all over the world.

References

1. Kroo, Ilan., Steve Altus, Robert Braun, Peter Gage, and Ian Sobieski, "Multidisciplinary Optimization Methods For Aircraft Preliminary Design," AIAA paper 94-4325-CP, 1994.
2. Ian Foster, Carl Kesselman, Jeffrey Nick and Steven Tuecke. , "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," 2002.2
<http://www.globus.org/research/papers/ogsa.pdf>.
3. Web Service Architecture, W3C Working Draft , 14 November 2002., URL:
<http://www.w3.org/TR/2002/WD-ws-arch-20021114/>.
4. Buning P, "Sources of Error in the Graphical Analysis of CFD Results", Journal of Scientific Computing, 1988 (3:2):149-164.
5. Darmfoal D and Haimes R, "An Analysis of 3D Particle Path Integration Algorithms", Journal of Computational Physics 123, 1996: 182-195.
6. Aid R and Levacher L, "Numerical Investigations on Global Error Estimation for Ordinary Differential Equations", Journal of Computation an Applied Mathematics, VOL82 (1-2), 1997:21-39.
7. Schroeder, W.J., Avila, L S and Hoffman, W., "Visualizing with VTK: a tutorial", IEEE Computer Graphics and Application. 2000, 20(5), 20.
8. William, J. S., Kenneth, M.M., Lisa, S.A., "The VTK user's guide", New York: Kitware, 2000: 1-355.

EEMAS: An Enabling Environment for Multidisciplinary Application Simulations

Lijun Xie^{1,2}, Yao Zheng^{1,2}, Jifa Zhang^{1,2}, Xin Huang^{1,2}, and Zhengge Huang^{1,2}

¹ College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China

² Center for Engineering and Scientific Computation, Zhejiang University,
Hangzhou, 310027, P.R. China

Abstract. EEMAS environment is a problem-solving environment for multidisciplinary application simulations. Within the EEMAS, there are four categories of modules involved, namely pre-processing module, computing module, post-processing module, and platform control module. The EEMAS is developed for complex and large-scale simulations to take advantage of powerful parallel and distributed computing technologies. All the modules are coupled through a software bus, which maintains the share memory and makes the modules integrated seamlessly. In the present paper, detailed design principles and applications of the EEMAS are addressed.

1 Introduction

A Problem Solving Environment (PSE) is a computer system that provides all the computational facilities necessary to solve a target class of problems [1]. Typically, it can reduce the difficulty of physical simulation by utilizing user natural languages and application specific terminology, and by automating many lower level computational tasks. We define a kind of PSEs in the following formula [1-3]:

PSE = User interface + Enabling libraries and tools + Problem solvers + Software bus

Commonly, a PSE should have a friendly user interface such as nature language and graphical user interface that can help the user to use the system in a direct and efficient manner. Enabling libraries and tools are the most valuable parts of a PSE. They provide all the necessary assistant functions for a simulation, such as geometric modeling, mesh generation, and scientific visualization. Problem solvers are integrated into the computational module, for various problem fields. Software bus is the method to integrate all the modules to work seamlessly and efficiently.

This paper describes a software architecture and implementation of a problem solving environment EEMAS, which stands for Enabling Environment for Multidisciplinary Application Simulations. It incorporates many modules to support large-scale simulations. Its main functions cover pre- and post- processing, computational platform control and the method to integrate all these modules together. The environment can reduce the time required for problem definition and for post-processing,

whilst the unified environment is ideal for multidisciplinary design applications, as the data is handled in one consistent format. It hides many aspects of computational engineering that are not of prime interest or relevance to the engineer, e.g. the setting-up and subsequent execution of an application on a parallel platform. The EEMAS is designed mainly for large-scale simulations, and heavily depend on visual steering, parallel and distributed computation. It is designed for multi-disciplines such as aerospace engineering, civil engineering, energy engineering, geosciences and oceanography.

2 Software Architecture and Features

The EEMAS framework is developed with C++, kernel algorithms are implemented with C and Fortran, the graphical user interface is built with Qt, and visualization capabilities are developed based on the OpenGL library. The system runs on Linux/Unix platforms, while its front end is being ported to Windows at this moment. Fig. 1 is a snapshot of an EEMAS session running on an SGI IRIX machine.

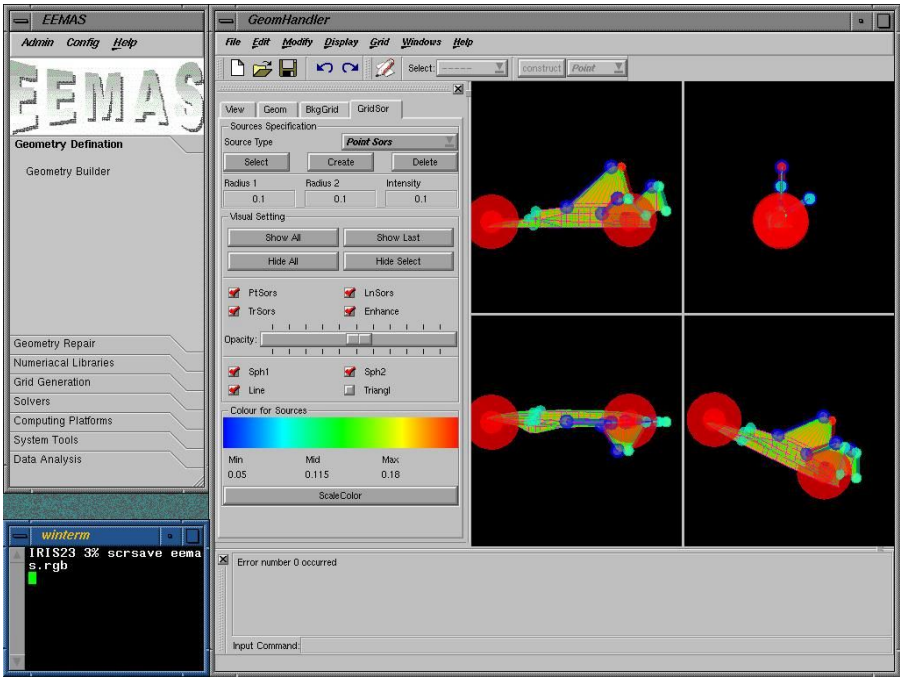


Fig. 1. A snapshot of an EEMAS session running on an SGI IRIX machine

2.1 Design Goals

The primary task of the EEMAS is to provide an efficient environment that enables scientists and engineers to create multidisciplinary application simulations, to develop

new algorithms, and to couple existing algorithms with powerful enabling tools. The main design principles and goals that guide development in the EEMAS project are as follows.

Abundant Functions. The EEMAS contains generic modules that provide necessary functions needed in mesh-based simulations, such as geometric modeling, CAD repair, mesh generation, domain decomposition, scientific visualization, platform control, and numerical libraries.

Scalability and Seamless Integration. As a Problem Solving Environment (PSE), the main aim of the EEMAS project is not to provide concrete scientific computational functions, but to provide an efficient, flexible and yet consistent framework, that facilitates integration of domain solvers and enabling tools. For this purpose, much attention is paid to the scalability of the system, which is mainly implemented with consistent data format and flexible data transfer interface.

The major portion of data is stored in share memory to reduce the consumption of memory and to improve the speed of data transfer. Three data transfer schemes are provided, that is, through pipes, sockets and temp files, respectively. For a module with its source code, users or developers are able to integrate it into the system by means of data transferring through pipes or sockets, of which the efficiency is quite high. If the source code is not available or the users do not want to spend much time on the integration, temporary files can be used for data transferring directly.

Visual Steering and GUI. The EEMAS follows the philosophy of visual steering. Its graphical user interface guides users to utilize particular components without in-depth expertise on those components, and to control the computing in a straightforward way. This allows scientists to use visualization tools while focusing on computational algorithms, and allows programmers to create visualizations tools without implementing simulation modules either. Visualization and numerical feedback are used throughout the system.

Parallel and Distributed Computing. The EEMAS is designed mainly for large-scale simulations. Therefore, majority of its modules utilize parallel and distributed computing, and can run on remote machines. Most of the modules, from mesh generation to problem solving, and to visualization, can run in the parallel mode. A module for parallel environment setup and control is also developed. Two parallel schemes are utilized. One is task parallelism that distributes different parts of a simulation to different processors. The other is data parallelism that is more widely used within a computationally intensive module.

2.2 Architecture and Functions

Fig. 2 describes the EEMAS architecture from users' perspective. There are four categories of modules involved, named as pre-processing module, computing module, post-processing module and platform control module. The first three are common phases of a simulation whilst the last one serves for the entire process of simulation. All the modules are coupled through a software bus, which maintains the share memory and makes the modules cooperate seamlessly.

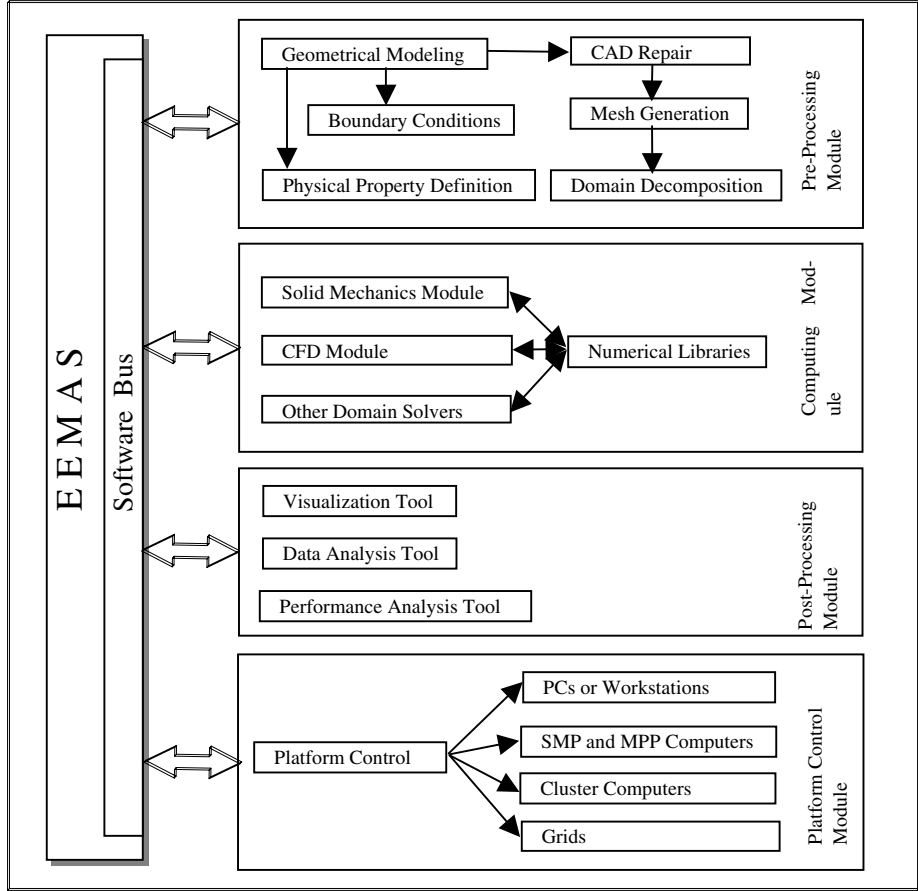


Fig. 2. The EEMAS architecture from users' perspective

2.2.1 Pre-processing Module

Pre-processing module deals with the problem definition and preparation for computing. This is the most time-consuming part in simulations, and many researchers are focusing on automating this work. In the EEMAS, the pre-processing module consists of a basic geometrical handling tool, a powerful mesh generators, and tools for general CAD format conversion, boundary condition definition and physical properties definition.

The geometrical handling tool in the EEMAS stores geometrical data by means of boundary representation. It is not as powerful as commercial CAD software, but it is adequate to process the common modeling work, especially with certain functionalities oriented to mesh generation. For complicated geometries, the user can directly model them with other CAD software, and then import them into the EEMAS.

Mesh generation is a crucial step in simulation that impacts both the calculating time and the accuracy. Research on mesh generation technologies is challenging,

while its importance is evident. The EEMAS provides powerful mesh generators with the ability to generate 2D planar meshes, surface meshes of triangles, and volume meshes of tetrahedrons. This module is capable to generate high quality meshes by means of visual steering. Users can specify the mesh spacing in terms of a background mesh and mesh sources.

Mesh generation has a stricter requirement of validation to geometrical models than common CAD software does. It is often the case that the format of geometries inside CAD packages does not meet the requirement for mesh generation. Holes, gaps, overlaps, and intersection of surfaces must be dealt with before mesh generators start. However, issues on geometry validity are not well defined, and it is still a large research field today. In the EEMAS, a geometrical validation and repair module is integrated, so as to find out many common cases of invalid models.

2.2.2 Computing Module

As mentioned above, the goal of designing the EEMAS environment is to support multidisciplinary application simulations. In general, scientists and engineers could integrate various domain solvers into the EEMAS to process particular simulations. To integrate with the EEMAS, users should adapt their solvers with the EEMAS data structure, or write an interface to convert formats. Data transfer involved can utilize pipes, sockets or temporary files. At present, an FEM solver for solid and structure analyses has been integrated.

2.2.3 Post-processing Module

Resulting data of complex and large-scale simulations are often difficult or even impossible to be understood without the assistance of visualization. Two powerful visualization packages named OpenDX [4] and ParaView [5] have been integrated into the EEMAS.

In order to support visualization for large datasets, such as gigabyte datasets visualization, parallel and distributed visualization technologies are employed. There are two modes of distributed visualization. One is the task parallel mode, in which different part of visualization pipes are processed by different processors. The other is data parallel model, in which the data is broken into pieces to be processed by multiple processors. The second method is easier to be implemented, because many visualization algorithms need not much change when running in parallel. The EEMAS supports both distributed and local rendering, and their combination. This provides scalable rendering for large data sets without sacrificing performance when working with smaller data sets.

For a parallel program, programmers and users usually have to tune the parameters and methods repeatedly to get a good performance. Thus a performance analysis tool is necessary. The EEMAS uses a tool named ParaGraph to visualize the behavior and performance of parallel programs on message-passing parallel computers. ParaGraph provides several distinct visual perspectives from which to view processor utilization, communication traffic, and other performance data in an attempt to gain insights [6].

2.2.4 Platform Control Module

The EEMAS provides a platform control module to process the setup of parallel environments, computing source management, file transfer and so on. This hides lot of trivial details of platforms from users, and helps users utilize all kinds of computers from local PCs. The whole system only needs to be initialized once. At present, this module can be used to explore local or remote Unix/Linux systems running on personal computers, workstations, SMP and MPP supercomputers, and clusters. The functions to utilize grid resources are also under development [7].

3 Applications

As an example, a structure analysis of a crank is processed within the EEMAS. After geometrical modeling, the crank is decomposed into 16 domains and delivered to different processors for parallel execution, and at last the data is visualized. We conduct this simulation on an SGI Onyx 3900 supercomputer and a Dawning PC cluster with diverse parameters to explore the performance of the system.

Fig. 3 shows the corresponding geometrical model, discretization of the model and the simulation result. The simulation has been carried out with various numbers of elements and those of CPUs on both the supercomputer and the PC cluster. Fig. 4 shows the efficiency for cases with the numbers of elements and CPUs. From the figure we can see that parallel processing is capable to reduce the computing time greatly. However, more CPUs cannot always guarantee higher performance. When too many CPUs are utilized, domain decomposition, data transfer and other communication operations will take lots of time and reduce the performance.

4 Conclusions and Future Work

The EEMAS is a problem-solving environment for multidisciplinary application simulations. It is a framework that can easily integrate arbitrary modules, and it contains various enabling tools such as pre-processing, post-processing, platform control and so on. It is especially designed for complex, large-scale simulations to take advantage of powerful parallel and distributed computing technologies.

The major work in the future is to extend its applications, by integrating more domain solvers to deal with special simulations. At present, an FEM solver for solid and structure analyses has been integrated, and the inclusion of a fluid dynamics solver is underway. Other modules, such as stereo distributed visualization and platform control tool, are also being improved.

Acknowledgements

The authors wish to thank the National Natural Science Foundation of China for the National Science Fund for Distinguished Young Scholars under grant Number 60225009.

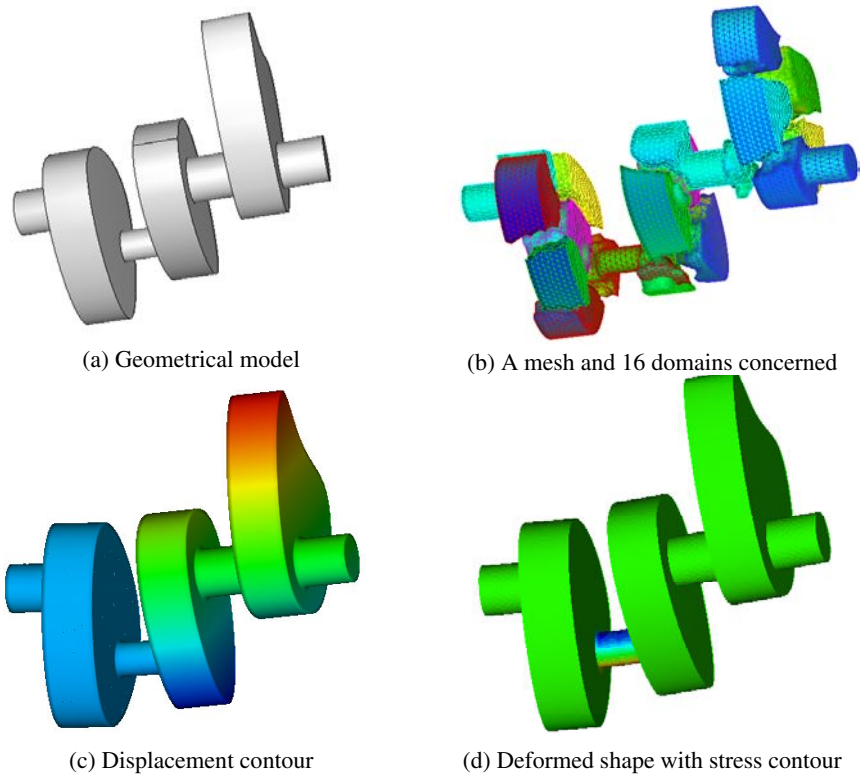


Fig. 3. Simulation of a crank

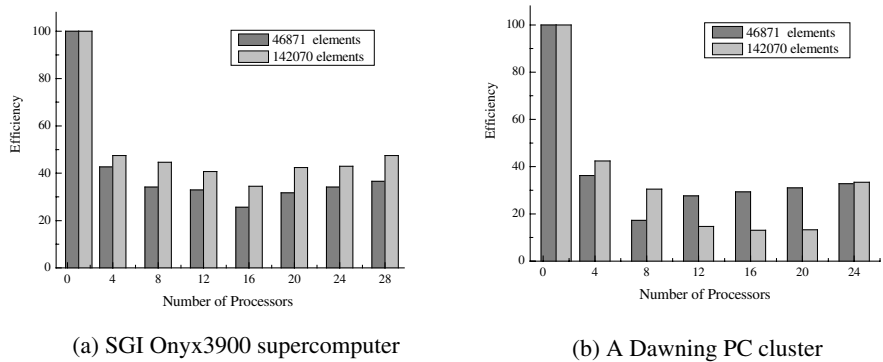


Fig. 4. Efficiency for cases with various numbers of elements and those of CPUs

References

1. Gallopoulos, E., Houstis, E.N., Rice, J.R.: Future Research Directions in Problem Solving Environments. Technical Report, Department of Computer Sciences, Purdue University (1992) 92-132
2. Rice, J.R., Boisvert, R.F.: From scientific software libraries to problem-solving environments, *IEEE Computational Science and Engineering*, 3 (3) (1996) 44-53
3. Rice, J.R.: Future Challenges for Scientific Simulation. Chapter 27 in *Enabling Technologies for Computational Science: Frameworks, Middleware and Environments*, The Kluwer International Series in Engineering and Computer Science, Vol. 548, Houstis, E. N., Rice, J.R., Gallopoulos, E., and Bramley, R., eds., Kluwer Academic Publishers, Boston (2000) 7-19
4. OpenDX User's Guide. URL: <http://www.opendx.org/started.html> (Current January 1, 2004)
5. Paraview User's Guide. URL: <http://www.paraview.org/HTML/Features.html> (Current January, 2004)
6. Heath, M.T.: Recent developments and case studies in performance visualization using ParaGraph. In Haring, G. and Kotsis, G., editors, *Performance Measurement and Visualization of Parallel Systems*, Elsevier Science Publishers, Amsterdam (1993) 175-200
7. Wei, G., Zheng, Y., Zhang, J.: Grid Service-Based Parallel Finite Element Analysis, *Proceedings of the Second International Workshop on Grid and Cooperative Computing (GCC2003)* (Shanghai, China, 2003), *Grid and Cooperative Computing: Second International Workshop, GCC 2003, Part I, Lecture Notes in Computer Science*, Vol. 3032, (Li, M., et al, eds.), Springer-Verlag, Heidelberg (2004) 123-130

Services for Parallel Remote-Sensing Image Processing Based on Computational Grid

Xue-Jun Yang, Zhi-ming Chang, Haifang Zhou, Xiangli Qu, and Chun-Jiang Li

School of Computer Science, National University of Defense Technology, ChangSha
cazimi@163.com

Abstract. The great advancement in remote-sensing technologies has brought new challenge to remote-sensing image processing: remote-sensing image processing demands processing capabilities of larger scale and cooperation in broader scope. Computational Grid provides rich computational resources and powerful storage capacity, which enables the sharing and cooperation within large scope and offers an ideal platform for remote-sensing image processing. In this paper, the parallel remote-sensing image processing software: PRIPS is encapsulated into a kind of Grid service in computational Grid. In this way, the service system for parallel remote-sensing image processing: PRIPSS-G is implemented. We first introduce the architecture of the parallel remote-sensing image processing software: PRIPS, then present its service implementation in Grid, and finally give some operational interfaces of this system and some related experimental results.

1 Introduction

With the rapid innovations of information and remote-sensing technologies, the spatial, radioactive, spectral and temporal resolution of satellitic remote-sensing images have been greatly advance. Data collected from satellite have multiplied 100-400 times than ever before ^[1]. Therefore, remote-sensing image processing comes to meet some new challenges, which includes:

First, with each passing day, the data that attained by means of remote-sensing have increased steadily, so the mass storage is one of the main problems that it confronts. At present, a remote-sensing image occupies tens of MB, even hundreds of MB. Even using compression encoding, for example, an image with size of 10000×10000 needs a hundred MB storage space to store. However, the ideal time in transmitting a normal image from a satellite is 2 seconds, it requires at least several TB storage space to store all the images a satellite takes in a day.

Second, because of the great amount of computation and the complex operation, remote-sensing image processing needs the capability of more large-scale computing. For example, for an image with a size of 10000×10000 , geometry correction in remote-sensing image preprocessing phase needs tens of billions of float multiplication and addition; other processing, such as automatic matching, segmentation, image classification of high spectrum and so on, computation is enormous due to massive image data.

Third, fast remote-sensing image processing technology is of urgent need in many fields, and parallel processing is one of the effective methods to solve that problem [2]. Applications in many fields, such as object reorganization and landform matching in strategics, weather forecast in meteorology, resource detection and navigation in geography, etc, urgently require fast remote-sensing image processing, but traditional software can't meet the high-performance demands of remote-sensing image processing any more.

Computational Grid enables the coupling and coordinated use of geographically distributed resources for such purposes as large-scale computation, distributed data analysis, and remote visualization [3]. It can solve these new problems that remote-sensing image processing does confront with now.

Grid technologies can integrate rich computation and storage resources that are provided by the Computational Grid, for this reason, the grid's aggregated computation and storage capacities are tremendous, and it can support distributed super computing in larger scale.

Second, Computational Grid can support traditional parallel applications. For example, using MPI-G2, a standard MPI program can run in Computational Grid, while the nodes running the programs could be Mainframe, Cluster, or heterogeneous computers that locate faraway, but all these are transparent to users.

Third, encapsulating remote-sensing image processing into Grid service can facilitate more users; therefore the value of software will be fully utilized. Additionally, Grid technologies can offer secure authentication mechanisms, provide dynamic and static information of software and hardware, and construct virtual labs for users.

With these requirements for the application of remote-sensing image processing in ChinaGrid which will provide with common services of "211 Project" of China ministry of education in the period of the Tenth Five-year Plan, we design and implement parallel remote-sensing image processing service system based on Computational Grid. The implementation of this system includes two steps: first, it needs program parallel algorithms for remote-sensing image processing; second, it needs encapsulate these parallel algorithms into services in Grid circumstances.

This paper is organized as followed: In the 2nd part, we introduce parallel remote-sensing image processing system (PRIPS); and next present the architecture of parallel remote-sensing image processing service system based on computational Grid (PRIPSS-G) in detail. The 4th part shows the interfaces and results of this system. Finally, we summarize the whole paper and discuss future work.

2 Parallel Remote-Sensing Image Processing System

According to the requirements mentioned above, parallel remote-sensing image processing is not only a problem urgently needed to solve, but also a necessary step bridge toward Grid application. But much more research is to be studied in remote-sensing image processing algorithms, and business software has done little on parallelism. Considering these factors, we develop PRIPS (Parallel Remote-sensing Image Processing System) software system, which covers many parallel algorithms we have studied to solve common problems of remote-sensing image processing. The PRIPS

system is designed as the idea of modularization and hierarchy, which makes the system maintainable and scalable.

2.1 Layered Structure of PRIPS

The system is composed of three layers as shown in figure 1, which are Common Module Layer, Image Processing Layer and Interface Layer.

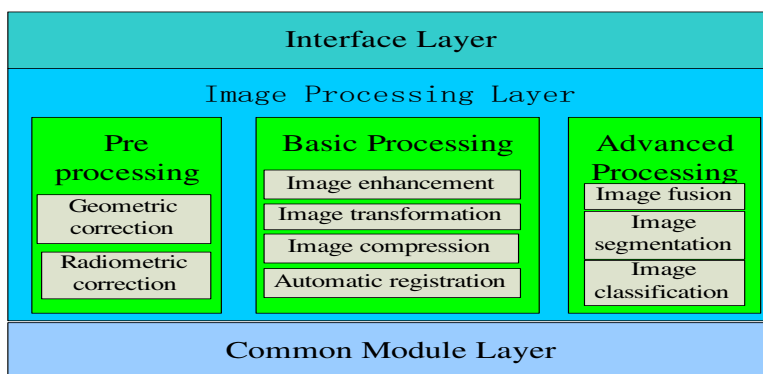


Fig. 1. The architectural of PRIPS software system

Common Module Layer is a common library, in which the functions are the fundamental part of remote-sensing image processing. It mainly consists of the transformation of image formats, access of image, basic segmentation of image.

Image Processing Layer consists of many parallel algorithms we have studied and some other algorithms existed. This layer is the core of the system and covers the three processing phases of remote-sensing image processing: preprocessing phase, basic processing phase and advanced processing phase.

According to the three processing phases, Interface Layer encapsulates these parallel algorithms into three interfaces in C++. Therefore, users can utilize these interfaces to construct their own applications of remote-sensing image processing.

2.2 The Modules in Image Processing Layer

Preprocessing phase: ①the module of geometric correction is used to correct a source image that has geometry aberration. This module provides with PIWA-LOC algorithm^[4], whose idea is that: first, management node divides a source image properly and sends these subimages to all computing nodes. Second, each computing node processes its own wrapping image and sends its processed image to the management node. Last, the management node stitches all the wrapping subimages to the target image. PIWA-LOC algorithm couldn't only solve the problem of data locality, but also could solve the problem of load balance for the transformation that don't wrap seriously, and is adapt to the nature of geometry correct. ②The module of radiometric correction uses to correct a source image that has radiation aberration caused by sensi-

tive changes of the sensor. This module provides with an algorithm that can correct aberration caused by height angles of the sun.

Base processing phase: ①the module of image enhancement uses to process a source image and produce a target image which has a better vision. This module includes space transformation enhancement (gray transform, histogram transform), space filter enhancement (low, median and high filter), frequency filter enhancement (low, high and middle filter). ②The module of image transformation uses to transform a source image from one domain to another domain. This module provides with some orthogonal transformations, such as FFT, DFT. ③The module of image compression includes loss compression and lossless compression. This module includes lossless compression such as RLC and JPEG-LS algorithms and loss compression such as wavelet-based loss EZW algorithm; also corresponding uncompressed tools should be given. ④The module of automatic registration is to make two images match closest for gray and space. This module provides with an algorithm called WAGR which is based on wavelet and can automatic matching during the whole processing [12]. The idea of WAGR is as followed: first, the source image is decomposed to a serial of images that have different precision and different size each other. Second, in the small size layer, we can get the best evaluation through linear search or other strategies, thus can evaluate the centre of the next size layer of the images. So the parameters could be more exact and refine by and by in the method. Finally, we can get the best results in the highest layer.

Advanced processing phase: ①The module of image fusion put images of high space and high spectrum into one image which could have more detail in the information of space and spectrum, so the image can be recognized easily. This module provides with two gray concretized algorithms which are HIS transformation and PCA transformation. ②the module of image segmentation is to segment a source image into several parts; Each part has its own character while other parts haven't, so can pick up the targets people are interested in. This module uses watershed segmentation algorithm [15] that has some advantages, one of which is that each segmentation part is close. This is very useful for other post processing, such as pattern reorganization. ③The module of image classification implements monitor classification and none monitor classification for the images of high space and high spectrum. This module provides with monitor classification, such as nearest neighbor and structural natural network and simple Bayes algorithms, and provides none monitor classification, such as ISODATA algorithm [16].

3 Parallel Remote-Sensing Image Processing Service System Based on Grid

Parallel Remote-sensing Image Processing Service System based on Computational Grid (PRIPSS-G) puts PRIPS software system into Grid environment, and encapsulates many remote-sensing image processing algorithms into Grid services to provide to users in wide area network. With these services, users can develop remote-sensing image processing applications that satisfy their own requirements.

3.1 The Architectural of PRIPSS-G

According to the criterion of Web Services, PRIPSS-G encapsulates algorithms of parallel remote-sensing image processing into services. The design of the system adopts a layered architecture as shown in figure 2, which is detailed as follows.



Fig. 2. The architectural of PRIPSS-G system

3.2 The Layers of PRIPSS-G

Service Application Layer provides users with the mechanism of retrieving all the services the system offered. Users can retrieve all the Grid services of parallel remote-sensing image processing through a Web browser. In this layer, we list all the WSDL that specifies relevant service to users. These services can be registered to some UDDI centers, so that Grid users can get specification about the location and the interface of the service. We also implement Service Application Layer through Web Portal at present. With friendly interfaces of the portal, one can use all kinds of these algorithmic services, dynamically retrieve the status of the task submitted and examine the information of the task queue or utilization of HW resources.

Service Interfacing Layer implements all the invocation interfaces of these algorithmic services, which can map the invocation from a user into some specific algorithm. In this layer, we encapsulate all the algorithms of parallel remote-sensing image processing into the interfaces using JAVA. Consider that efficiency is the primary; standard C++ language and MPI library are adopted in the implementation of parallel programs. Therefore, we wrap up the algorithms to the interfaces according to the criterion of Web Services, thus platform could be independent of the program for the algorithms.

Grid-enabled Algorithms Layer implements all the algorithms of parallel remote-sensing image processing in the Grid environment. This layer uses MPICH-G2 [7] library, which is the standard implementation based on MPIv1.1 and performs MPI programs through Globus which can couple with various architectural computers. Therefore, all the algorithms of parallel remote-sensing image processing in the PRIPS could be performed directly in Grid environment using MPI-G2 by compiling again. In this layer, we implement the on-demand scheduling algorithm. Aiming at users' need, Grid resources are chosen and the granularity of parallelism is determined. In this way, the dynamic requirement of applications is satisfied.

The Grid Resource Layer provides all the Grid resources available to these algorithms, which include Grid software resources and Grid hardware resources. The software resources we used are Globus Toolkit™ [8] which is the middleware of Grid,

and the related systematic software above it, including Cog, MPICH-G2 and so on. The hardware resources we used can connect multiply computer resources (including a MPP and an eight-node Cluster) on our campus. Constructed in this way, the Grid resource layer can coordinate these computational resources to perform the concrete algorithm of parallel remote-sensing image processing.

4 Implementation and Result of PRIPSS-G

The PRIPSS-G system offers friendly interface to authenticated users. Via simple operations, users can submit the tasks of remote-sensing image processing and the system will perform corresponding algorithm of the service. To get the processing results a service offers, what users need to do is only to choose the basic services of their tasks and input the parameters the algorithm requires. The system can support many users to access simultaneously, and adopts the FIFO strategy as the scheduling algorithm. At any moment, users can get the information of the task queue and the state of the task which the user submitted on the portal, and can check the utilization of the hardware resources. This part will show the interfaces and results of the algorithm of watershed segmentation as an example.

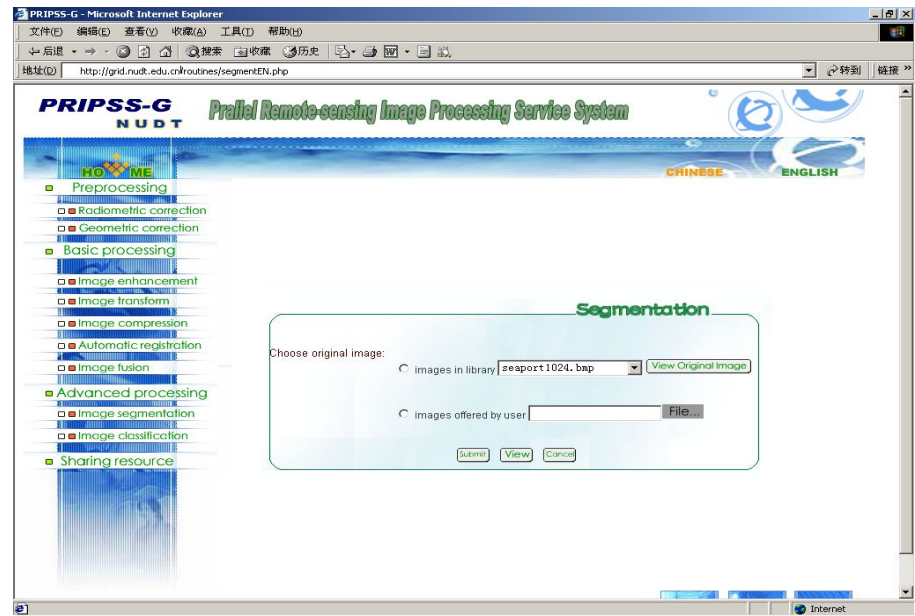


Fig. 3. The interface of watershed segmentation in PRIPSS-G system

On the left side of figure 3 are the basic services that the system provides with; the parameters of watershed segmentation which users need input are on the right side. These parameters include a source image (local images or users' uploaded images) and a smooth factor. The rough WSDL of this service is described as followed:

<definitions> WaterShedSegmentService
<types> the input parameters of waterShedSegmentRequest is string srcImageName and int smoothFactor;
the output result of waterShedSegmentResponse is int result
<message> waterShedSegmentRequest and waterShedSegmentResponse
<portType> the operation is PostProcessing::waterShedSegment, which is made up of response/request services
<binding> the protocol is SOAP HTTP
<service> http://grid.nudt.edu.cn/soap/servlet/rporouter is the location of the service

The visual result of watershed segmentation is shown in figure 4; the left image is the source image while the target image is on the right. These two images appear in an abbreviated form in the Portal browser. Users can download the processed images on their own.

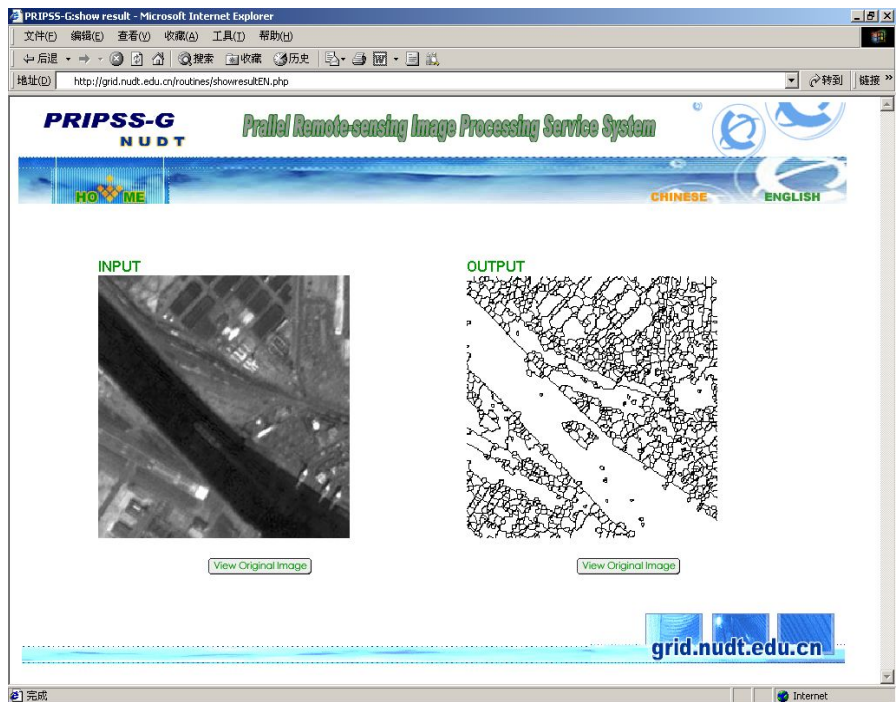


Fig. 4. The result of watershed segmentation in PRIPSS-G system

Graphic operation also avoids the complexity of invoking the Grid API and provides users with friendly interfaces. According to the experimental results, users can evaluate and analyze any specific algorithm.

5 Conclusion

In this paper, some complicated problems confronted remote-sensing image processing and the application needs of our project are summarized, and the major needs from remote-sensing image processing are tackled with the application of Grid. As a solution, we develop the PRIPS software to carry out parallel processing for remote-sensing image, which has speeded the whole processing. Each processing module of this software and the algorithms mostly adopted are illustrated in this paper. Our final solution is combining the PRIPS software with computational Grid and encapsulating it as a Grid Service, based on which the PRIPSS-G system is implemented. The architecture of PRIPSS-G system and the essential functionalities of each layer are demonstrated, and the basic interfaces and some processing results are presented in the end.

Yet, no matter the PRIPS software or the Web Portal design of PRIPSS-G system needs further improvement. All awaits our solutions in the near future.

References

1. S.L Zhu, Z.K Zhang. Gain and Analysis of Remote Sensing Image. Science Press. Beijing, China. (2000)
2. H.F Zhou. Study and Implementation of Parallel algorithms for Remote Sensing Image Processing, School of Computer Science NUDT. Chasha, China. (2003) 4-6 46-54
3. I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Int'l J. High-Performance Computing Applications*, Vol. 15, No. 3, (2001) 200-222
4. Y.H Jiang, X.J Yang, H.D Dai, H.Z Yi. A Distributed Parallel Resampling Algorithm for Large Images. *LNCS 2834/APPT2003*, Xiamen, China. (2003) 608-618
5. H.F Zhou, Y.H Jiang, X.J Yang. An enhanced parallel watershed algorithm based on components graphs for image segmentation. *Journal of Computer Research and Development*. Beijing, China. (2002)
6. Y.H Jiang. High-Accuracy and Parallel Algorithms for supervised Remote-sensing Image Classification, School of Computer Science NUDT. Chasha, China. (2004) 77-85
7. Nicholas T. Karonis, Brian Toonen, Ian Foster. MPICH_G2: A Grid-Enabled Implementation of the Message Passing Interface. In *Journal of Parallel and Distributed Computing (JPDC)*, Vol 63, No. 5, (2003) 551-563
8. Globus project. <http://www.globus.org/>.

Configuration of the Galaxy Grid Node Environment*

Guoteng Pan, Qiang Dou, Luoguo Xie, Guangming Liu,
Haifang Zhou, Xiaoqian Zhu, and Erhua He

School of Computer
National University of Defense Technology
Changsha, 410073, Hunan, P.R. China
tulip_1977@nudt.edu.cn

Abstract. With the demand of the applications increasing, today's supercomputers are not powerful enough to perform realistically large simulations. The emergence of grid can satisfy this need for computational power. In this paper, we first introduce the definition of the grid, analyze the characteristics of the computational grid, and outline grid computing research projects in China. In particular, we describe the architecture of the Galaxy grid node computing environment which contains the Galaxy computer system, front end server and grid system software. We also introduce the development of grid-based applications services.

1 Introduction

Over the past decades, the performance of the computer improves quickly, but the demand of the applications improves faster, today's applications are driving the requirement for higher-performance, larger-number resources. Many scientific applications have performed on dedicated supercomputer, but a supercomputer can not simulate a very large size problem in expected time. So "Grid computing" is proposed as a solution to this problem, it is intended to give users an easy and seamless access to remote resource. The term "grid" is defined as technologies and infrastructure that enable coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations (VO) [1]. A computational grid is collections of shared resources customized to the needs of their users, e.g., clusters, powerful supercomputers, collections of workstations, etc. This sharing is highly controlled, with resource providers and a consumer defining clearly, any of the authorized users within the VO have access to all or partial of these resources, and is able to submit jobs to the grid and expect responses.

Grid computing has grown rapidly since it emerged, more and more people and research institutions focus on the field. Some computer manufacturers have announced projects in grid computing such as IBM [2], HP [3], and Sun [4].

* This work is supported by the National High-Tech 863 Project of China under grant No.2002AA104510, and the Grid Project sponsored by China ministry of education under the grant No. CG2003-GA00103.

The paper is organized as follows. In Section 2, a brief overview is given for the China national grid project. The architecture of the Galaxy grid node computing environment is presented in Section 3. In Section 4, we introduce the development of two applications services based on the Galaxy grid node. Finally, we conclude the paper in Section 5.

2 China National Grid Project

In 1999, the Ministry of Science and Technology (MOST) launched the first computational grid project in China [5, 6]. In 2002, MOST launched the China National Grid Project (CNGrid), to build a China Grid system from year 2002 to year 2005. The architecture of the China Grid is shown in Figure 1.

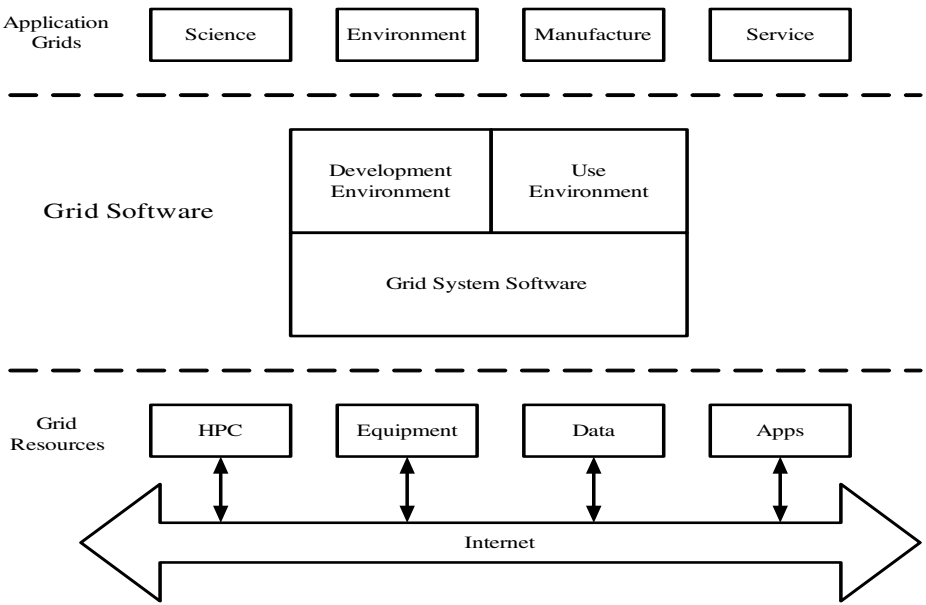


Fig. 1. Architecture of China Grid

3 Galaxy Grid Node Environment

3.1 Architecture

We started the research of grid computing based on the Galaxy computer in 2002 that is a node of the China grid. The key elements of Galaxy grid node computing environment (GGNCE) are the Galaxy computer system which contains 64 microprocessors, 32 GB memory and 1TB disks and can provide performance of 1000G FLOPS and communication bandwidth of 1.2GB/s, front end server and grid software running on the server. The architecture of GGNCE is shown in Figure 2.

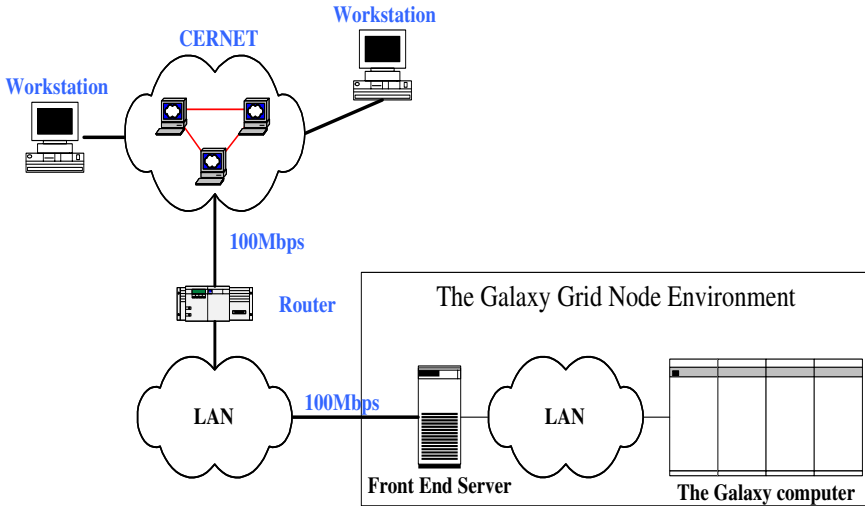


Fig. 2. Architecture overview of the Galaxy Grid Node Computing Environment, CERNET means China Education and Research Network

3.2 Function of Front End Server

The front end server in the GGNCE is a high-performance workstation that contains two network cards in order to connect to internet and intranet separately. The server's function is following:

- Separate the Galaxy computer from Internet.
- Running the grid system software called Vega GOS for supporting the grid computing.
- Receive and deal with remote user's request from outside, dispatch and submit computational task and return results to the user.
- Local user also can access and use the grid resources provided by the GGNCE through the server.
- Other security mechanisms.

3.3 Grid System Software

The grid system software running on the front end server is Vega GOS[8, 9] that was developed by Institute of Computing Technology (ICT), Chinese Academy of Sciences, it is a part of grid software platform layer, using OGSA/GT3 and web services as its basis. Vega GOS includes three layers by logistic division: the bottom of Vega device layer, providing the support for grid resources, the middle of Vega bus layer, managing the resource information and the top of Vega operation environment (VOE) layer, providing the support for user environment. The relation between them is showed in Figure 3.

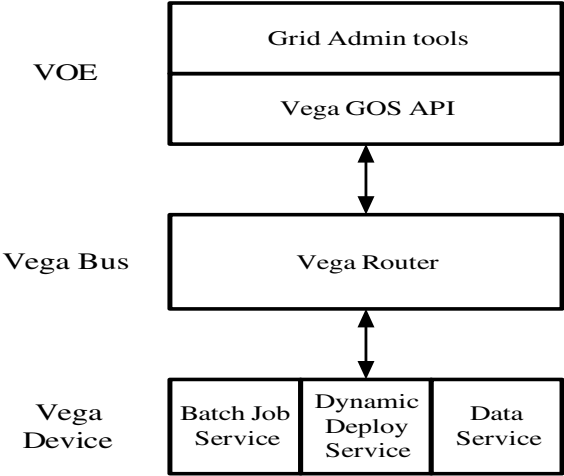


Fig. 3. Relation between Three Layers of Vega GOS

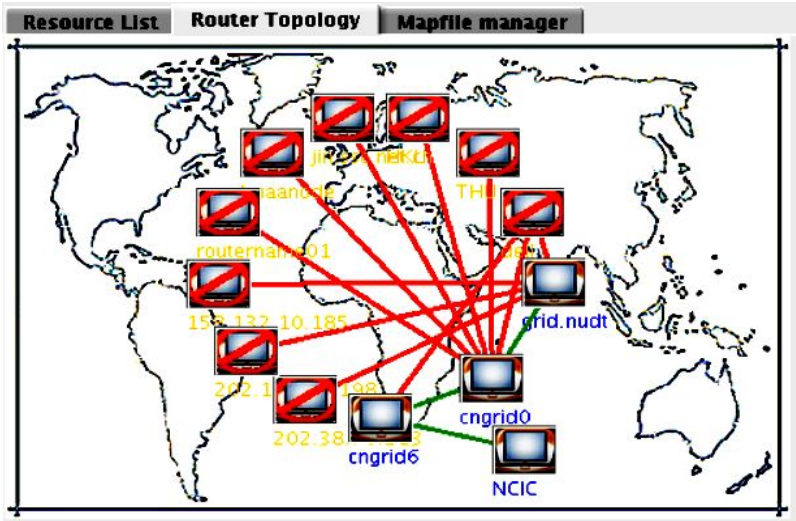


Fig. 4. Example of the Router Topology of the CNGrid

After installing and configuring Vega GOS properly, we can see the grid router topology of CNGrid (Figure 4) from Vega bus admin tool (VBA) of the software.

4 Development of Grid-Based Applications Services

4.1 Experimental Grid Service

With the development of the technology for satellite remote sensing, the resolution of the remote sensing image becomes higher and higher and the size of the image also

becomes larger and larger. On the other hand, the modern applications demand the processing speed of the remote sensing image faster and faster, the single computer can not keep these challenges for large images. So we develop an experimental grid service according to the new algorithms for remote sensing image processing which presented in [7].

The development contains five steps as follows:

1. Create the interface of the service: A grid service advertises its capabilities via a well-defined remote interface.
2. Write the implementation of the service: It is separated from its definition.
3. Write the deployment descriptor.
4. Build the grid service, creating a gar package: Package our configuration, schemas and codes into a gar package.
5. Deploy the gar package into the Vega GOS.

After launching VOE, our grid service appears in the resources list (Figure 5).

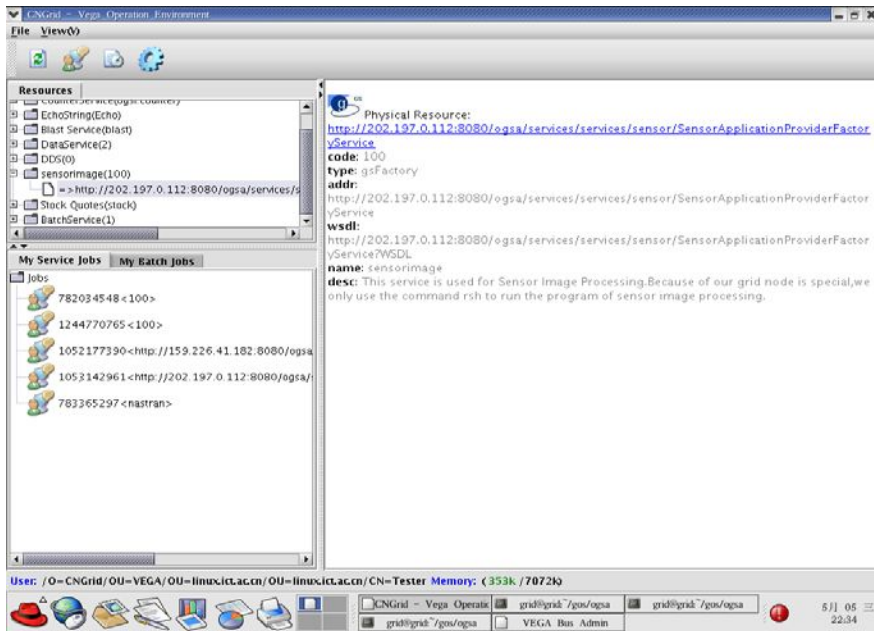


Fig. 5. Resources List in the VOE

4.2 Mesoscale Numerical Weather Predication

Mesoscale aerography is an important branch of modern meteorology. Mesoscale models, with grid resolution higher than global models, and with advanced physical parameterizations, have been an important tool for meteorological research over the past twenty years. Because of the extent of the computation required, meteorologists have invariably required the biggest and fastest computers to do their numerical modeling, grid computing is a good choice for satisfying this need.

We have implemented a high resolution mesoscale numerical weather prediction system[10] on the Galaxy grid node, that includes global medium-range weather forecast, limited regional model, the explanation and application of numerical weather forecast product and five-dimensional visualization.

We also have developed a web service for the mesoscale numerical weather prediction, a user can customize the schemes and parameters of prediction that he likes from the web, and he must do five steps through web before submitting job. Figure 6 is a sample web page, in this web page, user can customize physical parameters for some schemes, such as explicit schemes, cumulus parameterization schemes, planetary boundary layer parameterization schemes, radiation schemes and shallow convection schemes.

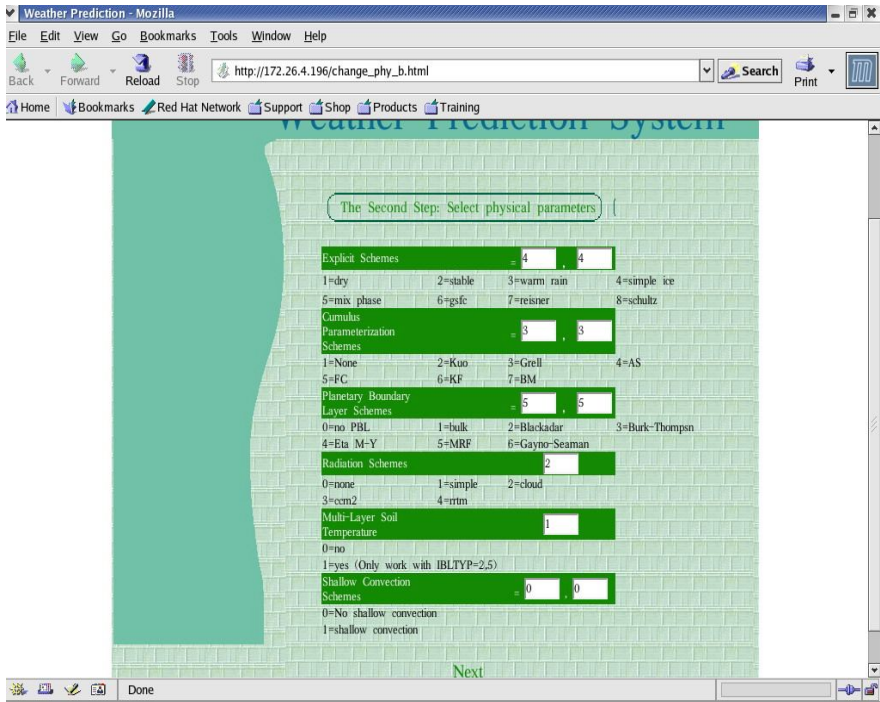


Fig. 6. Web Page of Selection the Physical Parameters for Schemes

When a user finishes the first four steps, he can submit the job from web, after the computation of the weather prediction complete, the result of prediction will return back to the user. If he is interested in the temperature and the humidity of the former selected region, he will get two pictures from web page show as Figure 7.

5 Conclusion

Because of the characteristics of some applications and their requirement for higher computational power, single computer can not satisfy this need. In this paper, we

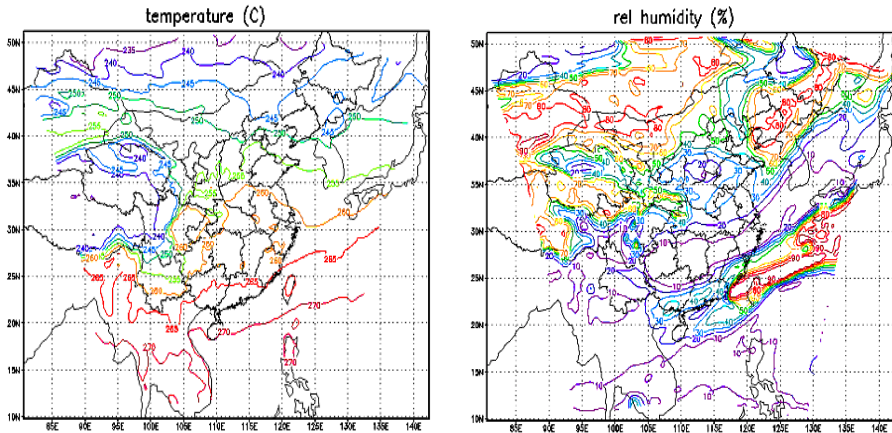


Fig. 7. Experimental Forecast for the Temperature and Humidity

introduced the definition of the grid, outlined grid computing research projects in China and particularly described the architecture of the Galaxy grid node computing environment. We also introduce the development of grid-based applications such as mesoscale numerical weather predication and an experimental grid service for remote sensing image processing.

Reference

1. Foster and C. Kesselman (eds.). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Fransisco, CA, 1999
2. <http://www-1.ibm.com/grid/>
3. HPTC Grid Computing. <http://www.hp.com/techservers/grid/>
4. <http://www.sun.com/solutions/infrastructure/grid/>
5. Z. Xu, N. Sun, D. Meng, W. Li. Cluster and Grid Superservers: The Dawning Experience in China, Proc. of the 3rd IEEE Int'l Conf. on Cluster Computing, 2001, pp. 351-358
6. Zhiwei Xu, Ninghui Sun, Wei Li, Taoying Liu. Dawning Superservers and the Vega Grid, HPC-Asia 2002, Keynote
7. Haifang Zhou. Study and Application of Parallel Algorithms for Remote Sensing Image Processing [PhD Dissertation]. National University of Defense Technology, Changsha, China, 2003
8. The Manual of Vega GOS. http://vega.ict.ac.cn/vega_gos_manual.pdf
9. The Development of Vega GOS. http://vega.ict.ac.cn/vega_gos_dev.pdf
10. Junqiang Song, Weimin Zhang, Xiaoqian Zhu et al, The Grid-computing Implementation of a Mesoscale Numerical Weather Forecast System. *Journal of Computer Research and Development*, 2002, 39(8): 968-972

GVis: A Java-Based Architecture for Grid Enabled Interactive Visualization*

Youbing Zhao, Wei Chen, Yingjun Qiu, and Jiaoying Shi

State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China
{zhaoyb, chenwei, soloqyj, jyshi}@cad.zju.edu.cn

Abstract. In this paper we present GVis – a Java-based software architecture for grid enabled interactive visualization. Compared with traditional parallel solutions that use multiprocessor computers or PC clusters, GVis provides a grid supporting environment that enables transparent conglomeration of heterogeneous resources, dynamic and autonomous coordination of visualization tasks and collaboration among end users. A portal is provided to end user for launching tasks and viewing results. With a Java-based object oriented visualization framework, the system can be extended and adapted conveniently to support a variety of visualization tasks.

1 Introduction

Visualization is an integral part of scientific computation and simulation [1, 2] and scientists today are increasingly relying on visualization for interrogation and analysis of synthesized or acquired data. Unfortunately, practical visualization tasks are resource demanding for CPU cycles, memory, and rendering capabilities.

Traditionally visualization tasks run in parallel on high-end multiprocessor computers or special-purpose rendering hardware [3, 4]. Recently with the fast advances of PC hardware and network technology, PC clusters emerged as an alternative at a much lower price [5].

Nonetheless the idle cycles, memory and rendering capabilities of PCs in organizations and on the Internet have not yet been fully exploited. In addition, it's difficult to access powerful multiprocessor computers and PC clusters through Internet.

The emerging grid technology [6, 7] seems to address these problems. Grid technology predicts that applications will run, communicate and share resources across the whole Internet. Despite that the technology itself is immature, grid has been regarded as a suitable solution to many resource demanding applications such as super computing and high performance visualization.

As type of motivating applications of the grid technology [6, 8], visualization has been on the way from traditional parallel and distributed solutions to the embracing of grid technology [9].

In this paper we present the on-going work of GVis: our grid enabled visualization architecture and system that enables coordinated and distributed visualization across multi-PCs. As of the writing of this paper, a prototype has been implemented.

* This work is partially supported by National 973 Program of China under grant No. 2002CB312105 and 2003CB716104.

The rest of this paper is organized as follows: Section 2 describes the related work. In Section 3, we explain the functionality and architecture of GVis in details. Preliminary experimental results are presented in Section 4, followed by conclusions and future work in Section 5.

2 Related Work

Resource-critical visualization tasks benefit greatly from parallel and distributed computing. These algorithms can be classified into two broad categories: image order and object order [10] or in other terms, sort-first and sort-last [11] modes. Detailed survey and classification of these parallel algorithms can be found in [12] and [13].

Kniss *et al.* implemented an interactive 3D texture-based volume rendering system on a high-end 128-CPU SGI Origin 2000 [14]. For the shared-memory architecture of the machine, no explicit network programming is required. It has a high performance I/O system for interactive visualization of time-varying T-bytes sized datasets.

Wylie *et al.* used PC clusters for object order parallel visualization [5]. MPI (Message Passing Interface) was used for network communication. A special active pixel encoded (APE) data structure was devised for compression of the color and depth buffers so that composition can be performed directly on compressed data.

Mahovsky proposed a Java-based software architecture for general-purpose distributed visualization on multi-PCs [15]. It uses a pixel based image parallel scheme and simple socket based client-server communication. This simple architecture achieves real-time frame rates at the expense of image fidelity.

Meanwhile, research work on grid enabled visualization has gone underway. Researchers in the Computational Visualization Center of University of Texas at Austin designed a grid enabled visualization system [16] based on their previous research on remote visualization [17]. Grid middleware Globus [18] is used for resource and data management and a grid portal is provided for access to a set of dedicated visualization servers.

Bethel *et al.* in LBNL (Lawrence Berkeley National Laboratory) have also done lots of work in grid enabled visualization. They developed an object order visualization backend VisaPult [19] for visualization of T-bytes sized scientific data. They made the system grid enabled by connecting it to grid middleware Cactus [20]. A noticeable contribution of their work is a high through-output data transfer scheme using connectionless UDP protocols. In addition, a web based VisPortal [21] is developed to enable access to a rich set of visualization tools (VisaPult included).

Compared with those grid enabled visualization systems developed or under development, our system focuses more on the integrated system architecture and the construction of a grid supporting environment for general-purpose resource, task and user management.

3 System Architecture

3.1 Design Goals

As described earlier, substantial idle resources (computing cycles, memory, storage, etc) are not fully utilized and many high-end equipments can not be easily accessed

outside of modern enterprises and organizations. Our GVis architecture and system attempt to exploit these resources in a more standard, collective, convenient and efficient way:

The design goals of our system can be summarized as 3Cs:

Conglomeration

By conglomeration, we imply that GVis provides *transparent* access to a large collection of *heterogeneous* resources. Besides traditional high-end hardware, our system can also support a large variety of existing desktop computers.

Coordination

As a multi-user multi-task environment that provides access to a variety of heterogeneous resources for a variety of tasks, the system must provide *dynamic* and *autonomous* management of resources, tasks and users. These functionalities are denoted collectively as “Coordination”.

Collaboration

Collaboration means that end users of GVis can collaborate, cooperate and share resources through the coordination of GVis system. In terms of the Grid, GVis supports *Virtual Organization (VO)* [22] seamlessly.

3.2 System Overview

A brief overview of GVis system is shown in Fig. 1. The whole system consists of three parts: GVis Portal (GVPort), GVis Supporting Environment (GVSE) and GVis Visualization Framework (GVVF).

The left box represents the client side – **GVPort** – the interface between end users and the GVis System. Visualization tasks can be launched through GVPort and the result is displayed interactively in Presenter.

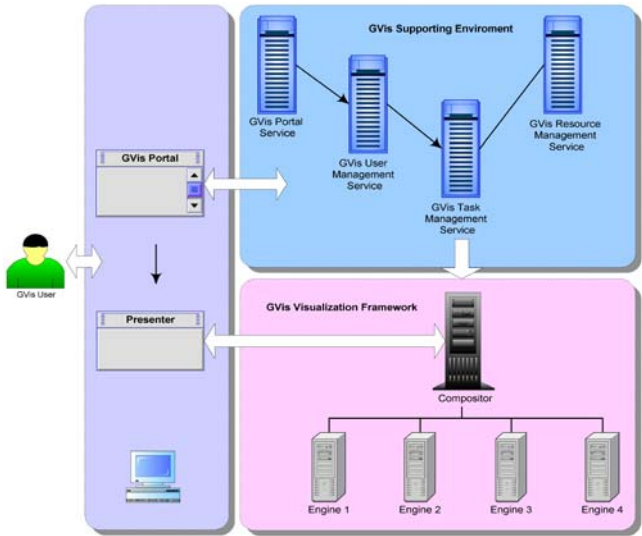


Fig. 1. Overview of GVis system

The top right box GVSE is responsible for grid related functionalities of resource, task and user management. It's the foundation of GVis architecture.

The bottom right box GVVF is a relatively independent distributed visualization framework based on GVSE and responsible for distributed execution of various visualization tasks. Presenter in the client side also belongs to GVVF.

To use GVis system the users need to "login" through GVPort first. After a successful login, GVis User Management Services is invoked to create a User Proxy Service for that user. Subsequent requests from the user will be transmitted to the User Proxy Service directly. For example, when starting a visualization task, task related information is sent to User Proxy Service to start a new specific Task Management Service (TMS). Subsequently TMS consults the Resource Management Service to acquire sufficient resources to run the task. In case of a successful resource allocation, TMS will start a Compositor and several rendering Engines. And each Engine renders a sub region of the screen or dataset. Final results are composed and blended by the Compositor and sent back to the Presenter.

3.3 Detailed Architecture

The overall architecture of GVis is shown in Fig. 2. Our design of GVis is based on Java 2 platform 1.4 and Globus 3. The foundation of GVis – GVSE – is based on the data management, task management, information service and grid security architecture provided by Globus 3. GVSE provides supports to GVPort and GVVF. GVVF is based on JOGL (Sun's semi-official Java-OpenGL binding [23]) and has its own messaging backend. Both JOGL and the messaging backend rely heavily on the new NIO package provided by Java 2 version 1.4. We will discuss GVSE, GVVF and GVPort in details in the following subsections.

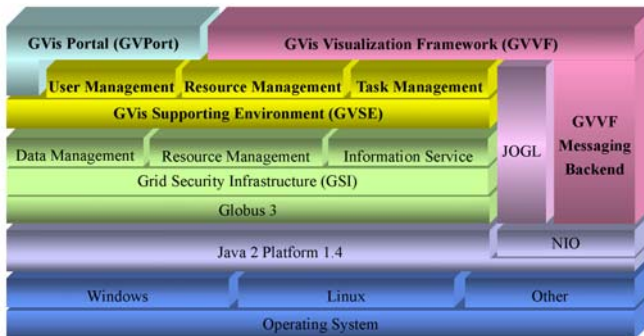


Fig. 2. Overall architecture of GVis

GVSE (GVis Supporting Environment)

GVSE is the foundation of GVis. It is based on grid middleware Globus, we chose Globus because it is the de facto standard grid middleware and provides low level supports such as grid security infrastructure, data and resource management and information services. GVSE utilizes capabilities provided by Globus to construct its customized user, resource and task management services. Basically it acts as a middleware layer between GVVF and the underlying Globus platform. The resource

management services are responsible for resource registering, monitoring, matching, allocation and reclaim. The task management services are responsible for launching, monitoring, scheduling and dependence resolving of tasks. The user management services are responsible for user authentication, per-user metering and collaboration among users. These three management services are the enabling services to the 3C-functionality described in 3.1.

GVVF (GVis Visualization Framework)

GVVF is responsible for the execution of visualization tasks. The design goal of GVVF is to provide a general-purpose distributed visualization framework. GVVF is not limited to some pre-defined visualization algorithms, instead it provides a framework that can be easily extended and adapted. The main components of GVVF are: a socket based messaging backend and three main abstract classes: Compositor, RenderEngine and Presenter (shown in Fig. 3).

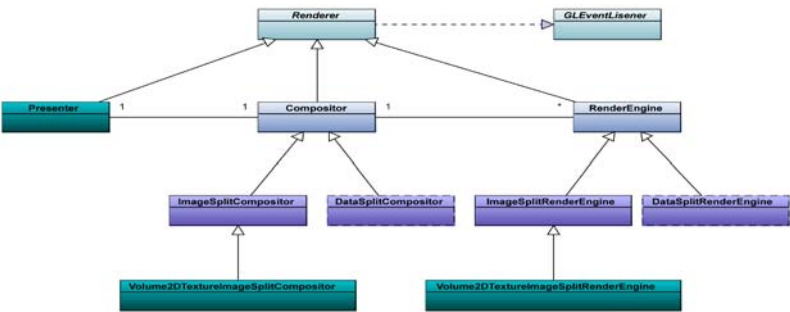


Fig. 3. A simplified class diagram of main classes in GVVF

GVVF is based on Java language and J2SE 1.4 platform. We chose Java for its cross-platform nature, simple programming model and strong thread support. The reason for choosing J2SE 1.4 is that it provides supports to direct buffer, selectable channel and full screen mode. As Java-OpenGL bindings are concerned, we chose JOGL for its semi-official background.

The underlying messaging backend of GVVF was developed from scratch using SocketChannel and ByteBuffer in J2SE 1.4. We chose to develop a backend by ourselves because we want to achieve simplicity, efficiency and maximum control of communication details.

Three main classes in GVVF, namely, Presenter, Compositor and RenderEngine, are inherited from abstract class Renderer which implements the GLEventListener interface of JOGL and contains basic information and functions needed for a rendering task. Compositor and RenderEngine are abstract classes from which image space splitting and object space splitting classes can be derived. Real rendering tasks are accomplished by concrete compositors and engines.

GVPort (GVis Portal)

Grid portals shield users from the underlying complexity of the grid. The standard view of grid portals is web-based system for accessing grid resources and launching

grid applications. Though web based grid portals inherit well both conceptions and software designs from web portal, the limitations of DHTML and Java applet make it difficult to implement full functional GUIs and applications in web browsers [21]. Different from web based grid portals, we envision grid portals act as “grid desktops” for computers “on the Grid”. Consequently, we prefer a full Java implementation instead of following the mainstream web based grid portal design [21, 24].

4 Preliminary Results

Our system has been partly implemented. An image space splitting compositor and rendering engine has been implemented for 2D texture volume rendering. Resource management services and task management services of GVSE have been partly implemented. We can start compositors and rendering engines through command line consoles in the client side, and the preliminary results show the feasibility and effectiveness of our prototype system. Detailed information about the testing environment is shown in Table 1 and results are listed in Table 2.

Table 1. Testing Environment

ID	Machine	OS	CPU	Mem	Display Card	FPS
0	CAD14	Redhat Linux 9.0	PII 350M	256M	WinFast 3D L2300V (8M)	0.2 – 0.4
1	CAD15	Debian GNU/Linux	PIII 600M	392M	Nvidia Riva TNT2 Pro (32M)	30-40
2	CAD177	Windows 2000	P4 1.7G	256M	Nvidia Geforce4 MX440 (64M)	30-40
3	CAD22	Windows 2000 Advanced Server	PIII 600M	256M	Nvidia Riva TNT2 Pro (32M)	25-35
4	LDEV	Debian GNU/Linux	PIII 800M	256M	Nvidia Geforce MX 400 (64M)	30-40
5	YONEX	Windows 2000	P4 1.7G	512M	Nvidia Geforce4 MX440 (64M)	30-40

Table 2. Test Results

Test	Presenter	Compositor	Engines	Canvas Size	FPS
1	0	2	1,3	288 x 270	3-4
2	0	2	1, 3, 4, 5	288 x 270	3-4

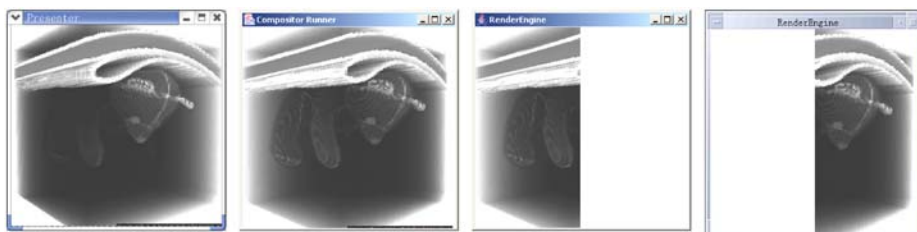


Fig. 4. Screenshots from test 1, taken from computer 0, 2, 3, 1 from left to right respectively

The data set used in the tests is a 128x128x64 Teddybear volume data. We used the 2D texture rendering engine shown in Fig. 3. The image resolution is 300x300. The FPS (frames per second) in Table 1 is acquired with the same data set and image resolution (300 x 300) with JOGL on a single machine. The compositor window and engine window of test one are shown in Fig. 4. From the results we can see that our system can render a moderate sized volume data at nearly interactive rates on low-end machines where interactivity can not be achieved before.

5 Conclusions and Future Work

The proposed GVis system has an integrated architecture which provides a full functional grid enabling environment, an extensible Java based visualization framework and a grid portal. The preliminary results show that our system, though not full-fledged, has brought relatively high-end visualization capabilities to low-end PCs. Future work to make a full-functional grid enabled interactive visualization system include: user management and task scheduling services of GVSE; implementation of GVPort to experiment the idea of “grid desktop”; implementation of object space splitting renderer and interactive transfer function design in Presenter; well-defined data management services to support dynamic data transfer.

Acknowledgements

Special thanks should be given to Mr. Mao Yingli (the chinese name of Mr. Rody Klein, a visiting scholar from University of Savoie, France) for his kind help and careful proofreading of this paper. We'd also like to thank Jian Yang, Zhefan Jin, Shengyou Lin, Fuli Wu, Zonghui Wang and Haoyu Peng for their kind help. In the end we'd like to thank VisualParadigm (<http://visual-paradigm.com>) and ConceptDraw (<http://www.conceptdraw.com>) for their wonderful trial version software.

References

1. Drebin R. A., Carpenter L., Hanrahan P.: Volume Rendering. *Computer Graphics*. 4 (1988) 65-74
2. Shi J.Y., Cai W.L.: *Algorithms and Systems of Scientific Visualization*, Science Press. 1995 (in Chinese)
3. Pfister H., Kaufman A.: Cube-4 – A Scalable Architecture for Real-Time Volume Rendering. *Symposium on Volume Visualization*. (1996) 47-54
4. Pfister H., Hardenbergh J., Knittel J., Lauer H., Seiler L.: The VolumePro Real-Time Ray-Casting System. *Proc. SIGGRAPH 1999*. (1999) 251-260
5. Wylie B., Pavlakos C., Lewis V., Moreland K.: Scalable Rendering on PC Clusters. *IEEE Computer Graphics & Applications*. 21(4) (2001) 62-70
6. Stevens R., Woodward P., DeFanti T., Catlett C.: From the I-WAY to the National Technology Grid. *Communications of the ACM*. 40(11) (1997) 50-60
7. Foster I., Kesselman C., Jeffrey M., Nick J., Tuecke S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Global Grid Forum*. June (2002)

8. DeFanti T., Foster I., Papka M., Stevens R., Kuhfuss T.: Overview of the I-WAY: Wide Area Visual Supercomputing. *Int. J. Supercomputer Applications and High Performance Computing*. 10(2) (1996) 123-130
9. *IEEE Computer Graphics & Applications*. 23(2) (2004)
10. Neumann U.: Communication Costs for Parallel Volume Rendering Algorithms. *IEEE Computer Graphics and Applications*. 14(4) (1994) 49-58
11. Molnar S., Cox M., Ellsworth D., Fuchs H.: A Sorting Classification of Parallel Rendering. *IEEE Computer Graphics and Applications*. 14(4) (1994) 23-32
12. Yagel R.: Towards Real Time Volume Rendering. *Proc. GRAPHICON'96*, Vol. 1. Saint-Petersburg Russia (1996) 230-241
13. Wittenbrink C.M.: Survey of Parallel Volume Rendering Algorithms. *Proc. of Int. Conf. on Parallel Distributed Processing Techniques and Applications*. (1998) 1329-1336
14. Kniss J., McCormick P., McPherson A., Ahrens J., Painter J., Keahey J., Hansen C.: Interactive Texture-Based Volume Rendering for Large Data Sets. *IEEE Computer Graphics and Applications*. 21(4) (2001) 52-61
15. Mahovsky J., Benedicenti L.: An Architecture for Java-Based Real-Time Distributed Visualization. *IEEE Transactions on Visualization and Computer Graphics*. 9(4) (2003) 570-579
16. Center for Computational Visualization of University of Texas at Austin. Grid Enabled Visualization. Poster of NPACI AHM. (2003)
17. Bajaj C., Park S., Thane A.G.: Parallel Multi-PC Volume Rendering System. CS & ICES Technical Report, University of Texas at Austin. (2002)
<http://www.ices.utexas.edu/CCV/papers/Smpark-VolVis02.pdf>
18. Globus: <http://www.globus.org>
19. Bethel E.W., Shalf J.: Cactus and Visapult: An Ultra-High Performance Grid-Distributed Visualization Architecture Using Connectionless Protocols. *IEEE Computer Graphics and Applications*. 23(2) (2003) 51-59
20. Cactus: <http://www.cactuscode.org>
21. Siegerist C., Shetty P., Shalf J., Bethel W., Jankun-Kelly T.J., Kreylos O., Ma K.L.: Vis-Portal: Deploying Grid-Enabled Visualization Tools Through A Web-Portal Interface. Wide Area Collaborative Environments Workshop. Seattle, WA. (2003)
22. Foster I., Kesselman C., Tuecke S., The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *Int. J. Supercomputer Applications*. 15(3) (2001)
23. JOGL: <https://jogl.dev.java.net/>
24. Grid Sphere: <http://www.gridsphere.org>

Balancing CPU and GPU: Real-Time Visualization of Large Scale 3D Scanning Models

Zhao Dong, Wei Chen, Long Zhang, and Qunsheng Peng

State Key Laboratory of CAD&CG, Zhejiang University, 310027, Hangzhou
{flycooler, chenwei, lzhang, peng}@cad.zju.edu.cn
<http://www.cad.zju.edu.cn>

Abstract. Recent advances in 3D scanning technologies have enabled us to acquire large scale point-clouds data rapidly. *Point-based representation* has been introduced as a versatile and powerful graphics primitive. This paper proposes an adaptive rendering algorithm for large scale point models. The algorithm first subdivide the target model into multiple *patches* in the preprocess. A hierarchical structure is built for each patch and then converted into a linear binary tree. During rendering, the model is processed patch by patch. Fast visibility decision is made to cull invisible patches. Visible patches are displayed in graphics processing units (GPU) by choosing appropriate rendering mode, i.e, a distance-dependent strategy. Our algorithm takes full advantage of GPU and effectively balances the workload between CPU and GPU. We also propose a fast compression/decompression technique which achieves 8 times compression ratio. Experimental results demonstrates high performance and image quality rendering for large scale 3D scanning models in consumer PCs.

1 Motivation

Point-based surface models define a 3D surface by a dense set of sample points captured by 3D scanning devices. The idea of using points as surface primitives was first proposed in 1985 [1]. Since then, point-based graphics has been paid more and more attentions. The Qsplat system [2, 3] integrates a multi-resolution hierarchical structure into the rendering of point models. Though it achieves interactive frame rates by performing view frustum and back-face culling for each selected hierarchical sub-tree, Level-of-Detail (LOD) selection on-the-fly requires significant CPU computation while reducing the image quality. The idea of Sequential Point Tree (SPT) [4] converts the hierarchical structure into linear buffers, facilitating a graphics hardware implementation which improves the efficiency dramatically.

The point-based rendering algorithms mentioned above focus on efficiency and speed. But, few of them supports high quality rendering for models with complex surface textures. Pfister *et al.* [5] proposed to represent each point with a well-defined Surfel. The set of Surfels constitute a water-tight surface. Zwicker *et al.* [6] introduced the elliptical weighted average (EWA) [7] resampling filter



Fig. 1. From left to right: Buddha, 1.06M points, 13.06FPS; Dragon, 1.28M points, 11.77FPS; Lucy, 10.07M points, 10.25FPS.

to overcome the aliasing arisen from perspective projection. However, a non-optimized software implementation of EWA surface splatting algorithm suffers low speed. Ren *et al.* [8] introduced object space EWA filter that can be efficiently implemented as a quad on modern graphics processing units (GPUs). Kobbelt *et al.* [9] further accelerated this method by representing Gaussian filter using Pointsprite primitive. Nevertheless, all these algorithms do not make use of hierarchical structure and thus can not be applied to large scale point models.

To overcome these limitations, we merge several points into one point and choose coarse rendering method in the regions far away from the viewpoint. For points near the viewpoint and around the silhouettes of models, precise and smooth rendering strategy is chosen. We design a well-defined adaptive rendering strategy to guarantee the smooth transition between coarse and precise rendering modes. Because GPU is a parallel streaming processor, it is designed as a co-processor of CPU in our algorithm. The selection of appropriate hierarchy and culling operations are accomplished efficiently in CPU. Meanwhile, valid points are handled in GPU, yielding well balance between CPU and GPU as well as high efficiency. Furthermore, a fast compression/decompression technique is proposed to place large scale models in video memory entirely.

2 Algorithm Overview

During the preprocess, the target model is divided into multiple patches, each of which represents a dense region of the surface. For each patch, individual hierarchical structure containing its bounding box and normal cone is established and converted into a linear binary tree. The linear binary tree enables convenient access in GPU which is a parallel streaming processor. During rendering, the model is processed patch by patch. Based on bounding box and normal cone of each patch, fast view frustum and back-face culling are carried out to discard invisible patches. This process is accomplished in CPU. Those visible patches

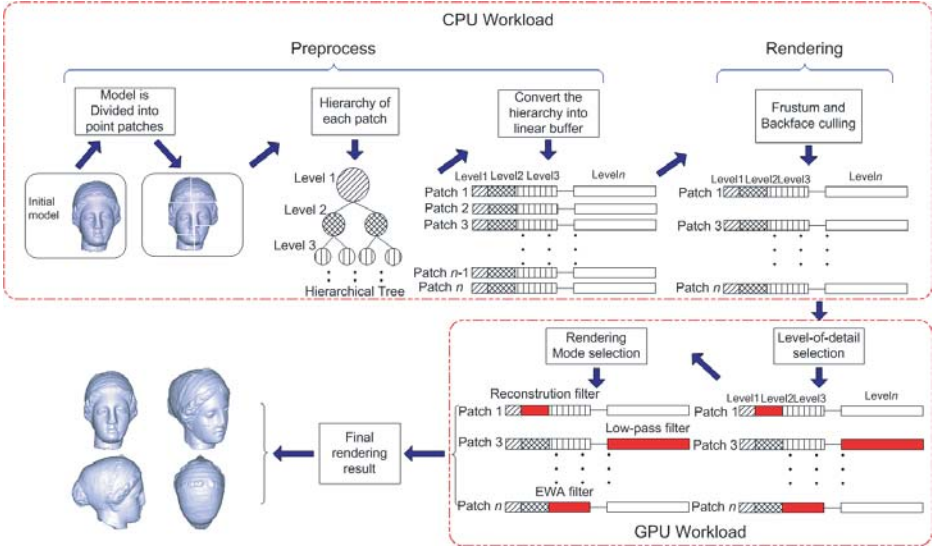


Fig. 2. Workload distribution between CPU and GPU for visualization of large scale point models.

are then handled in GPU by choosing appropriate rendering mode and view-dependent Level-of-Detail. Thus balancing between CPU and GPU is achieved. The whole pipeline is shown in Fig.2.

3 The Point Patch Structure

Typically, raw data sets from 3D scanner include position, normal and radius *etc.* Subdivision of one model consists of four steps. First, a linear table containing all points is generated. Subsequently it is divided into multiple sub-trees which corresponds one point patch. Covariance analysis method [10] is applied to find optimal segmentations. For flat regions, less point patches are produced. The covariance analysis method is also used to estimate the curvature and normal cone of each sub-tree node. Thereafter, each leaf-node is converted into a linear binary tree. The pointers to each node and assistant information, such as the maximal and minimal radii of the node, are recorded at the same time. This guarantees that the most appropriate level of each linear sub-tree is selected during rendering. Obviously, the proposed strategy processes much less extra points than SPT method [4] which adopts a octree-based structure.

4 Distance-Dependent Rendering Strategy

In this section, we introduce a distance-dependent strategy for simplification of complicated EWA resampling filter without loss of image quality.

4.1 EWA Filter

EWA filter [7] eliminates the aliasing caused by perspective transformation through introducing the low-pass filter in 2D screen space. Let $\{P_k\}_{k=1}^n$ denotes the point sets of model, the EWA resampling function $g'_c(x)$ in 2D screen space is defined as the convolution of reconstruction filter $g_c(x)$ of point data and low-pass filter $h(x)$ in 2D screen space.

$$g'_c(x) = g_c(x) \otimes h(x) = \int_{R^2} g_c(\xi) h(x - \xi) d\xi \quad (1)$$

The 2D expression of EWA filter function is an elliptical Gaussian function $G_V(x)$:

$$G_V(x) = \frac{1}{2\pi|V|^{\frac{1}{2}}} e^{-\frac{1}{2}x^T V^{-1}x} \quad (2)$$

where V denotes variance matrix of Gaussian function. Let V_k^r denotes variance matrix of reconstruction filter and V_h denotes variance matrix of low-pass filter. Each of them is a diagonal matrix. The EWA resampling filter $\rho_k(x)$ of P_k is as follows:

$$\rho_k(x) = G_{J_k V_k^r J_k^T + V_h}(x - x_k) \quad (3)$$

where x_k denotes screen space coordinates of P_k with perspective projection transformation $\mathbf{M}$, and J_k denotes Jacobian Matrix of $\mathbf{M}$. Then $H_k = J_k V_k^r J_k^T + V_h$ is the variance matrix of $\rho_k(x)$.

4.2 Distance-Dependent EWA Filter

Note that EWA resampling filter is the convolution of the reconstruction filter and low-pass filter. If the model is close to the viewpoint, the reconstruction filter dominates. If the model is far away from the viewpoint, the reconstruction filter effect is small so that the EWA resampling filter can be replaced by the low-pass filter.

Let $C_\eta = \sqrt{2 \cdot \tan^2(\frac{fov}{2}) + 1}$, where fov is the view angle, and $S_h = \pi \cdot r_h^2$ denote the effective area in screen space of low-pass filter. The ratio of S_{proj} to S_h determines the proportions of the reconstruction filter and the low-pass filter in the EWA resampling filter. We propose following adaptive EWA filter strategy.

```

ChooseRenderMode Strategy
For each point patch of model
  begin
    if  $S_{Proj}/S_h > c_{max}$  then
      Rendering with the reconstruction filter,  $H_k = J_k V_k^r J_k^T$ ;
    elseif  $C_\eta S_{Proj}/S_h < c_{min}$  then
      Rendering with the low-pass filter,  $H_k = V_h$ ;
    else
      Rendering with EWA filter.
  end

```

Here, c_{min} and c_{max} are adjustable parameters for balancing the efficiency and quality.

5 Workload in CPU

5.1 Patch-Based Frustum and Back-face Culling

For each patch, its precomputed bounding box is first clipped against the view frustum, which verifies whether the patch is inside the view field or not. In addition, the average normal and normal cone of the patch is used to estimate whether it is visible.

5.2 Level-of-Detail Selection

We traverse the pre-built binary tree from top to bottom. For each visible patch, the projection area and minimal distance from its bounding box to the viewpoint are first computed. If either of them is small enough, the current patch is used. Two special cases are further considered. If the average curvature of the patch is larger than some threshold, or its normal cone is larger than some threshold, its child nodes are processed.

5.3 Rendering Mode Selection

Once the Level-of-Detail of one patch is decided, the adaptive filter strategy is utilized to choose optimal rendering mode. For the region that has large curvature or is around silhouettes, EWA resampling filter is used.

6 Data Compression in GPU

For large scale point models, the required memory consumption is large. To avoid frequent communication between video memory and host memory, it is necessary to keep all information in video memory. In this section, we propose a fast compression/decompression technique which achieves 8 times compression ratio (Table.1).

Table 1. Point data compression statistics.

Data type	Before compression (bits)	After compression (bits)	Final (bits)
Position	96	32	32
Normal	96	16	16
Tangent	96×2	2×2	4
Texture Coordinate	64	2	4
Radius	32	8	8
Total	480=60 Bytes	62 Bits	64=8 Bytes

6.1 Compression of Position

In the preprocess stage, the bounding box of the target model is uniformly discretized into $256 \times 256 \times 256$ space grids. The position of each grid can be substituted by the grid's index number in three coordinates axe, and it consumes

3 bytes. Furthermore, each grid is subdivided into $8 \times 8 \times 4$ sub-grids to locate the position of each point. The index numbers of each sub-grid consume 1 byte among which the indices of X,Y,Z axe occupy 3 bits, 3 bits and 2 bits. As a result, the bounding box is actually divided into $2048 \times 2048 \times 1024$ grids, and the center of each grid is used to encode the position of points contained in the grid.

6.2 Compression of Normal Vector

Three components of each normal can be represented by trigonometric functions of θ and φ in the unit spherical coordinates. The domains of θ and φ are averagely partitioned into 256 discrete values. The corresponding proximal θ and φ of each normal are computed, and then the index numbers in [0-255] of two angles can be obtained, which occupy 2 bytes. This normal compression strategy can represent $256 \times 256 = 65536$ kinds of different normal vectors, the rendering quality of which can be visually satisfied.

6.3 Compression of Tangent Vectors

Since the tangent vectors are cross perpendicular to the normal, its compression is straightforward. For the normal (n_x, n_y, n_z) , the tangent vector can be chosen from three cases: $(0, -n_z, n_y)$, $(n_z, 0, -n_x)$ and $(-n_y, n_x, 0)$. Therefore its storage is 2 bits.

6.4 Compression of Texture Coordinates

There exist four instances, $(0, 0)$ $(0, 1)$ $(1, 1)$ $(0, 1)$, for texture coordinates. We create a simple lookup table which contains four elements. And it costs 2 bits. We further encode the 4 bits of two tangent vectors and 2 bits of texture coordinates into 1 byte.

6.5 Compression of Radius

For the radii of all points, we design a statistics-based clustering method. First, we compute their distribution and choose 256 radius candidates to build a 256-entries lookup table for compression. For each radius, it is approximated by some element of the lookup table and the index number costs one byte in video memory. The total memory consumption of each point is 8 bytes.

7 Results

We implemented our algorithm with DirectX 9.0b. Performance was measured on one PC equipped with an AMD Athlon 2G CPU, 1GB RAM and an ATI 9800 Pro video card with 256MB video memory.

Table 2 shows algorithm performance in frame per second (fps) for different sized 3D scanning data sets¹ at the frame buffer resolution of 512×512 . The column of *numbers of rendered points* denotes the number of points in the selected level during rendering. The Lucy model selects the point sprite rendering model and all the others select adaptive EWA rendering model. Analyzing the data in Table 1, the rendering speed of adaptive EWA rendering algorithm reaches 4.5M points per second, which is 3 times as much as the reported speed [8]. Considering our hierarchical structure, the adaptive EWA algorithm, when there exists a moderate distance between the model and the view point, will be up to 18M points per second. Instanced by Lucy model, it is obvious that our adaptive algorithm can render 100M points per second if simple render mode is chosen.

Table 2. Performance statistics of the algorithm. Image resolution: 512×512 .

Model	Render Mode	Num. of Points	Num. of Rendered Points	FPS
Lucy(Fig.1)	Point Sprite	10.073M	3.071M	10.25 fps
Dragon(Fig.1)	Adaptive EWA	1.28M	0.42M	10.75 fps
Buddha(Fig.1)	Adaptive EWA	1.06M	0.31M	15.12 fps
Hip	Adaptive EWA	0.53M	0.15M	38.12 fps
Hand	Adaptive EWA	0.33M	0.11M	55.23 fps
Lion	Adaptive EWA	0.18M	0.07M	80.50 fps

8 Conclusion and Future Work

The contributions of this paper are threefold. First, we introduced a flexible point patch structure for point model. Second, we proposed a Distance-Dependent Rendering Strategy. Third, we proposed a fast compression/decompression technique and achieve 8 times compression ratio to store all relative information in video memory locally.

Due to the workload balance between CPU and GPU, our algorithm, with the steady enhancement of GPU's capability, will achieve higher efficiency and better quality. As future work is concerned, we want to further optimize our implementation and adapt it to upcoming new hardware features. We are planning to extend our approach to distributed network application and Grid computing is our preferred technology. We aim at establishing a grid-based distributed real-time visualization platform for large-scale virtual reality application.

Acknowledgements

This work is partially supported by National 973 Program of China under grant No. 2002CB312100 and 2003CB716104, and NSFC of China (60103017, 60303028), Zhejiang Provincial Natural Science Foundation of China (R603046).

¹ All data sets are generated by 3D scanning devices. They are provided by Cyberware Co. (<http://www.cyberware.com>) and stanford graphics lab (<http://graphics.cs.stanford.edu>).

References

1. M. Levoy and T. Whitted.: The use of points as display primitives. Technical report, CS Departement, university of North Carolina at Chapel Hill, January 1985.
2. S. Rusinkiewicz and M. Levoy.: QSplat: a multiresolution point rendering system for large meshes. In Proc. In Siggraph 2001 Symposium on Interactive 3D Graphics, 2001.
3. S. Rusinkiewicz and M. Levoy.: Streaming QSplat: A Viewer for Networked Visualization of Large, Dense Models. In Siggraph 2000 Conference Proceedings.
4. C. Dachsbacher, C. Vogelgsang, and M. Stamminger.: Sequential point trees. In Siggraph 2003 Conference Proceedings.
5. H. Pfister, M. Zwicker, J. van Baar, and M. Gross.: Surfels: Surface elements as rendering primitives. In Siggraph 2000 Conference Proceedings.
6. M. Zwicker, H. Pfister, J. van Baar, and M. Gross.: Surface splatting. In Siggraph 2001 Conference Proceedings.
7. P. S. Heckbert.: Fundamentals of Texture Mapping and Image Warping. Master's thesis, University of california at Berkley, 1989.
8. L. Ren, H. Pfister, and M. Zwicker.: Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. In Eurographics 2002 Conference Proceedings.
9. M. Botsch and L. Kobbelt.: High-Quality Point-Based Rendering on Modern GPUs. In Pacific graphics 2003 Conference Proceedings, 2003.
10. M. Pauly, M. Gross, and L. Kobbelt.: Efficient simplification of point-sampled surfaces. In Proc. IEEE Visualization 2002, 2002.

LRZB, a Hybrid Algorithm of Local Ray-Casting and Z-Buffering for Large Geometric Datasets

Xiaoxu Han¹, James Cremer², Zhihong Wang², and Jun Ni²

¹ Applied Mathematical and Computational Sciences

² Department of Computer Science, the University of Iowa, Iowa City, IA 52242, USA
xihan@math.uiowa.edu
{cremer, zhihwang, jni}@cs.uiowa.edu

Abstract. This paper presents a hybrid algorithm, LRZB, for real-time rendering of large complex scenes. The basic LRZB algorithm decomposes scenes into two sub-scenes, and uses z-buffering to render the low depth complexity set and ray casting to render the remainder. The basic LRZB works well for densely occluded scenes. Several techniques to enhance the ray casting component of basic LRZB algorithm are presented: lazy ray casting, object-oriented ray casting, selective lazy ray casting, and coherent octree traversal. Experiments demonstrate that LRZB can achieve 2-10 times speed-up for large complex scenes compared with conventional z-buffer rendering. LRZB is efficient, easy to implement, and amenable to parallelization, especially in distributed Grid environments.

1 Introduction

Although graphics software and hardware has made impressive recent progress, there remains a need for new algorithms for rendering complex scenes in real time. Most computer game, CAD design, and scientific virtual reality and visualizations systems use z- (or depth) buffer-based graphics cards in which the rendering cost grows linearly with the number of primitives in the scene, even when the vast majority of the primitives are not visible. It is desirable to have an *output sensitive* rendering algorithm in which the time complexity depends primarily on what is visible, and is only weakly dependent on the total number of primitives in the scene.

The time complexity of z-buffering is $O(|S|+|Q|)$, where $|S|$ is the number of scene primitives (polygons) and $|Q|$ is the number of projected pixels generated in the rasterization process. Z-buffering is very efficient for rendering scenes containing a modest number of primitives with large projections and not too many hidden primitives. For large complex scenes, especially those with high depth complexity, the large amount of time spent rasterizing and rendering hidden primitives can make z-buffering completely ineffective.

The time complexity of ray casting, assuming no spatial sorting, is $O(|S| \times |\hat{V}|)$, where $|\hat{V}|$ is the number of pixels in the rendered image. With spatial sorting using octrees, BSP trees, or similar approaches, the complexity becomes $O(\log|S| \times |\hat{V}|)$. With sorting, ray casting becomes a front-to-back rendering method in which it is unnecessary to examine every primitive when rendering a pixel, thus making ray

casting less dependent on the scene-depth complexity. Unfortunately, ray casting lacks the hardware support available for z-buffering.

In this paper, we present a new hybrid rendering method that takes advantage of good features of both z-buffering and ray casting. The basic algorithm divides a scene S into two sub-scenes, S_{near} and S_{far} . z-buffering algorithm is employed for S_{near} , the sub-scene that is near to the viewer, contains many of the most visible primitives, and is likely to be of low depth complexity. Ray-casting is then used for S_{far} , the sub-scene containing primitives relatively far from the viewer, and containing many hidden primitives. This approach is very effective on single processor machines, but is also amenable to parallel, distributed, and Grid computing environments.

Section 2 reviews related work. Section 3 presents the basic LRZB algorithm. Section 4 describes the lazy ray casting and several other techniques that significantly improve the basic LRZB algorithm. Section 5 summarizes the results and discusses ongoing and future work.

2 Related Work

Rendering methods for large datasets typically focus either on visibility culling or scene approximation. Visibility culling approaches attempt to remove invisible objects to decrease the rendering overhead. Beyond the basic and well-known back-face and view frustum culling techniques, a number of effective culling-based methods have been developed, including Teller's cell-based PVS algorithm [4,15], the Greene's Hierarchical z-buffer algorithm (HZB)[7], Zhang's Hierarchical Occlusion Maps (HOM)[18], and others [5,9,17]. Many of the methods require a great deal of preprocessing time. A comprehensive review can be found in Cohen-Or *et al* [3].

Scene approximation-based methods attempt to increase rendering efficiency by reducing the overall number of primitives (e.g. level of detail approaches such as [6]) or the complexity of the primitives (e.g. point based rendering methods [11,12,16]).

3 The Basic LRZB Algorithm

The basic LRZB algorithm consists of preprocessing and real-time rendering stages. The basic LRZB algorithm's flow chart is shown in Fig. 1. In the preprocessing stage, an octree is built to do spatial sorting that speeds up ray casting. Details of the particular octree building approach used can be found in Han[8].

There are five steps in the real-time stage: frustum culling, clipping-plane selection, local z-buffer rendering to generate a partially completed image, frame-buffer reading, and ray casting to complete the image. The frustum culling step is common to most real-time rendering algorithms. The second step selects a clipping plane, Z_{clip} , that partitions the scene into sub-scenes $S_{\text{near}} = \{p: p \geq Z_{\text{clip}}\}$ and $S_{\text{far}} = S - S_{\text{near}}$ (see Figure 2). S_{near} is expected to have low depth complexity but cover much of the final image. S_{far} is expected to have relatively large image complexity. After rendering S_{near} using z-buffering, the frame buffer is read to determine which (if any) pixels remain unfinished. Ray casting on S_{far} is then used to complete the unfinished pixels.

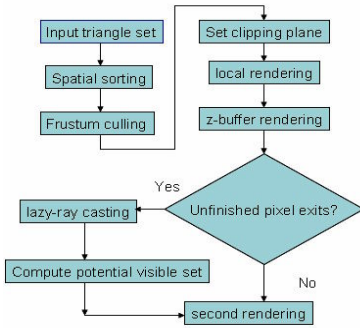


Fig. 1. The flow chart of naïve LRZB algorithm

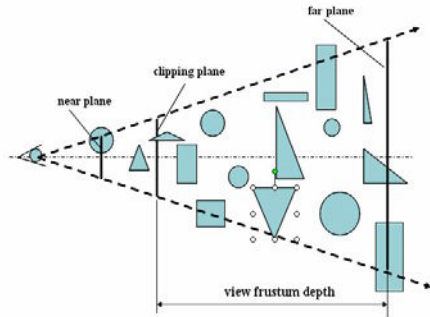


Fig. 2. Clipping plane

The pseudocode for the real-time stage is:

```
Real-time(Scene S, Eye E){
    Sculled = frustum-culling(S, E)
    (Zclip, Snear, Sfar) = determine-clipping-plane(Sculled, E)
    z-buffer-render(Snear, E)
    Punfinished = read-frame-buffer-to-determine-unfinished-pixels()
    for each pixel p in Punfinished do
        raycast(p, Sfar, E)
}
```

3.1 Determining the Clipping Plane

One of the interesting questions in the basic LRZB algorithm is how to determine a good clipping plane. A clipping plane is good if the depth complexity of S_{near} and the number of resulting unfinished pixels are both small. Figure 3 plots clipping plane distance versus resulting number of unfinished pixels for two different scenes.

An effective method for determining the clipping plane is to cast a small number of rays into the scene and then compute the average of the first-hit distances. Frame coherence methods can then be employed to compute clipping plane distances for subsequent frames starting from the distance of the previous frame.

3.2 Experimental Results for Basic LRZB

The basic LRZB algorithm was implemented and tested on a densely occluded “virtual city” scene (see Fig. 4) with 200,808 triangles and an average depth complexity of 9. On average, z-buffer-only rendering required 0.35 seconds, while LRZB rendering required 0.23 seconds. The percentage of time LRZB spent doing ray-casting increased with image resolution (Figure 5), due largely to the inefficient handling of empty/background pixels in the basic LRZB algorithm.

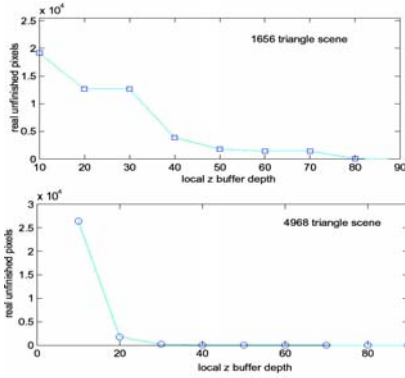


Fig. 3. Clipping plane moves closer to the near plane with the increase of scene primitive numbers

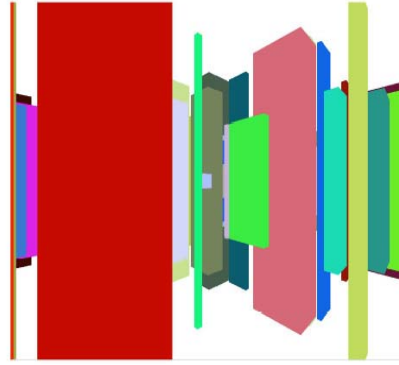


Fig. 4. A densely occluded scene with 200,808 triangles

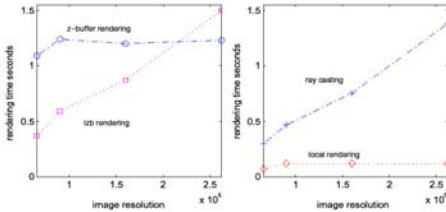


Fig. 5. Performance of basic LRZB

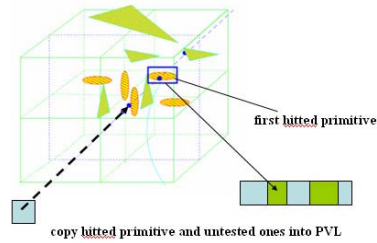


Fig. 6. Principle of lazy ray casting

4 Improvements to the Basic LRZB Algorithm

This section presents four techniques that improve the basic LRZB method: lazy ray casting to add hardware support to the ray casting stage, object oriented ray casting (OOR) to speed empty pixel processing, selective ray casting (SLR) to handle the worst case situations for ray casting, and image coherence-based octree traversal.

4.1 Lazy Ray Casting

Rendering via ray casting involves two basic tasks - nearest surface finding and shading – and can be relatively quite expensive. To improve the basic LRZB algorithm, we partition classic ray casting into two parts, using software ray casting to determine small sets of potentially visible primitives, and then taking advantage of graphics hardware to finish the rendering job. For each unfinished pixel, rays are cast simply to determine a conservative set of potentially visible primitives for that pixel. Primitives within an octree node are unsorted and so, in basic ray casting, each must be tested to

determine the closest surface. In the lazy ray casting approach, once it is determined that the ray intersects *some* primitive in an octree node, all the node's primitives are simply added into a global potentially visible list (PVL). After doing this for all unfinished pixels, the primitives in the PVL are rendered using z-buffering.

4.2 Object Oriented Ray-Casting

The basic LRZB does not work very well for some scenes whose final images have many empty or background pixels. In Figure 7, a virtual city of 572,412 triangles, more than 50% of the pixels are empty. For such scenes, a lot of time can be spent traversing long octree paths only to discover no ray-primitive intersections.

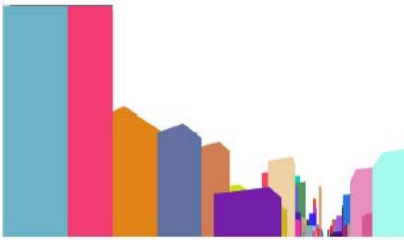


Fig. 7. A half densely occluded scene with 572,412 triangles

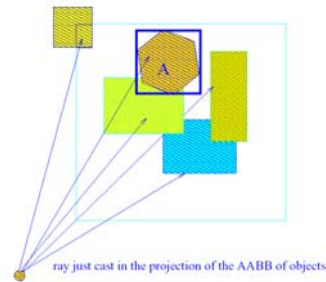


Fig. 8. Object-oriented ray casting (OOR)

The basic idea of object-oriented ray casting (OOR) is to use bounding box projections to quickly determine which unfinished pixels might yield ray-primitive intersections during ray casting. If, as is common, the many primitives of a scene are (or can be) grouped into higher-level *objects*, OOR is accomplished by projecting axis-aligned or oriented bounding boxes of each object onto the image plane and marking which unfinished pixels are covered. Another approach is to simply project the lower level non-empty nodes of the octree to the image plane. The addition of lazy and object-oriented ray casting to LRZB yields:

```

LRZB-with-lazy-and-object-orient-raycasting(Scene S, Eye E) {
    Sculled = frustum-culling(S, E)
    (Zclip, Snear, Sfar) = determine-clipping-plane(Sculled, E)
    z-buffer-render(Snear, E)
    Punfinished = read-frame-buffer-to-determine-unfinished-pixels()
    Pinteresting = compute-coverage-of-projections(objects(S), Punfinished)
    for each pixel p in Pinteresting do
        add lazyraycast(p, Sfar, E) to PVL
    z-buffer-render(PVL, E)
}

```

4.3 Selective Lazy Ray-Casting

The basic lazy ray casting approach of Section 4.1 can spend a lot of time searching for intersections with all the primitives associated with an octree node (the worst case

being when the ray misses all primitives in a fully populated node – see Fig. 9). The idea of selective lazy ray casting is to do ray intersection tests on only one or a few “local occluders” chosen from an octree node’s primitives.

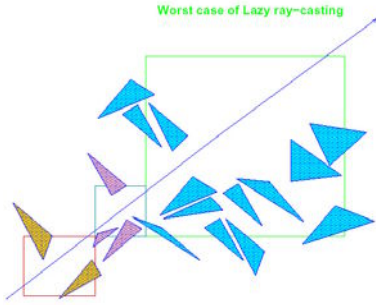


Fig. 9. The worst case in the lazy ray casting

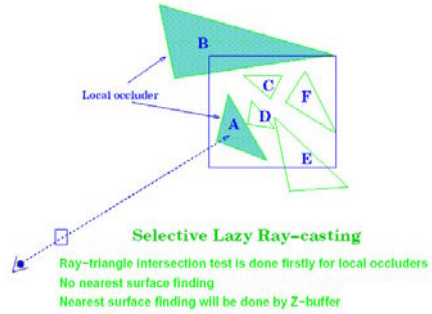


Fig. 10. Selective Lazy Ray-casting

A simple “static” approach selects, during octree construction, a node’s largest primitives as the local occluders.

At rendering time, a ray-octree node test begins with the node’s first local occluder. If the ray and primitive intersect, all of the node’s primitives are added to the global PVL, and the traversal for the given ray can terminate. If no intersection is found between the ray and local occluders, the nodes primitives are added to the PVL and the ray is tested against the next octree node along the ray. Compared with basic lazy ray casting, the use of local occluders decreases the number of intersection tests and increases the size of the PVL. Experiments exhibited average ray primitive test time reduction of 80% - 95% along with a factor of 1 to 5 increase in PVL size.

Static local occluder selection considers only each primitive’s size. A dynamic selection method that takes into account the orientation of primitives with respect to the eyepoint can be more effective. In particular, dynamic selection can be based on the $v \cdot n \cdot \text{area}(p)$, where v is the view direction, n is the primitive’s normal and $\text{area}(p)$ is the area of the primitive. Dynamic selection is obviously substantially more expensive than static selection, and not always worth the extra cost.

4.4 Ray Traversal Speed-Up

Several approaches exist for increasing octree traversal efficiency[1,2,10,14]. In the LRZB algorithm, spatial coherency can be exploited by noting that octree traversal paths of rays through neighboring pixels are likely to be similar. If a cached ray path is kept for the previous pixel, one can test if the first intersection for the next pixel’s ray is the same as in the cached ray path. If it is, an exit point is computed and the same query made for the next node in the cached ray path. When a test fails, a neighborhood search is done. Instead of employing a full but relatively costly approach such as Samet’s [13], one can easily do a partial neighborhood search involving just the node’s immediate siblings. If the sibling search fails, the ray path is ex-

tended via the standard octree traversal method. Experiments show that the coherence based octree traverse algorithm achieves, on average, 50% speed-up over the usual top down octree traverse algorithm. See Figure 11 for an example.

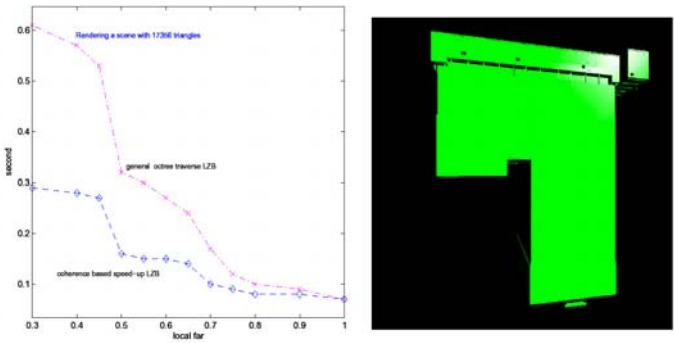


Fig. 11. An rendering example using coherence octree traverse

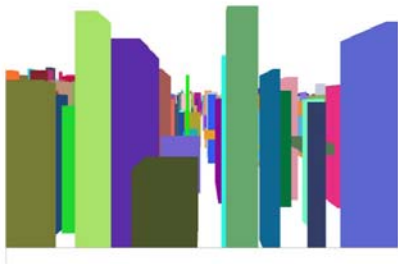


Fig. 12. A scene with 1,822,260 triangles

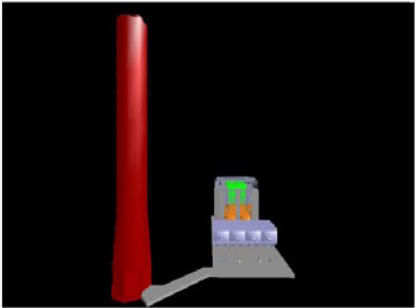


Fig. 13. UNC powerplant model

4.5 Rendering Experiments

Fig. 12’s 1,822,260 triangle virtual city scene required 3.14 seconds using standard z-buffering-only compared to 0.37 seconds using LRZB with the improvements of Section 4. Many additional tests were run on scenes using the 13 million polygon UNC power plant model (<http://www.cs.unc.edu/~geom/Powerplant/>) shown in Fig. 13. Statistics for these are presented in Han[8]. Experiments were conducted on a 1.8 MHz Pentium IV PC with 1G RAM and an Nvidia GeForce 3 series graphics card.

5 Conclusions and Future Work

Scenes continue to grow in complexity, so the need for new real-time rendering algorithms remains. As detailed in Sections 3 and 4, LRZB combines z-buffering and ray-casting methods to yield a very fast real-time approach for many scenes.

Several challenges and opportunities for future work exist. One of most promising is parallelism. In a parallel version of LRZB, n sub-octrees are built on n processors in the real time, hoping that substantial preprocessing time is saved under the parallel environment). The geometric model is partitioned, using bounding boxes, into n sections, and screen projections of all bounding boxes are computed on the root processor. Local rendering is also done in the root processor. Frame buffer reading results are then broadcast. Processor i receives the i th geometry section, the unfinished pixels in the projection of the i th section, and the sections whose projections are intersect with the i th's. Object-oriented lazy ray casting is done in each processor and the final image parts obtained are sent to the root processor.

References

1. Arvo, J.: Ray tracing with meta-hierarchies, SIGGRAPH 90 Advanced Topics in Ray Tracing Course Notes (1990)
2. Chang, A.: A survey of geometric data structures for ray tracing, Polytechnic University technical report TR-CIS-2001-06 (2001)
3. Cohen-Or, D. Chrysanthou, Y., Silva, C., Durand, F.: A survey of visibility for walk-through applications. IEEE Tr. Vis. and Comp. Graphics, 9(3), 412-431, 2003
4. Coorg, S., Teller, S.: Real-time occlusion culling for models with large occluders. ACM Symposium on Interactive 3D Graphics, p. 83-90 (1997)
5. Durand, F., Drettakis, G., Thollot, J. and Puech, C.: Conservative visibility preprocessing using extended projections, Proc. SIGGRAPH, p. 239-248 (2000)
6. Garland, M.: Quadric-based polygonal surface simplification, Ph.D. thesis, Technical Report CMU-CS-99-105, Carnegie Mellon University (1999)
7. Greene, N., Kass, M. and Miller, G.: Hierarchical z-buffer visibility, Proc. SIGGRAPH 93, p. 231-238 (1993)
8. Han, X.: The local z-buffer algorithm for rendering large complex scenes. Ph.D. thesis, Applied Mathematics and Computational Sci., The Univ. of Iowa (2004)
9. Klosowski, J. and Silva, C.: Efficient conservative visibility culling using the prioritized layered projection algorithm, IEEE Tr. Vis. and Comp. Graphics 7(2), 365-379 (2001)
10. Peng, Q., Zhu, Y. and Liang, Y.: Fast ray tracing algorithm using space indexing techniques, Proc. EUROGRAPHICS, 11-23 (1987)
11. Pfister, H., Zwicker, M., van Baar, J. and Gross, M.: Surfels: Surface elements as rendering primitives. Proc. SIGGRAPH 2000, p. 235-242 (2000)
12. Rusinkiewicz, S. and Levoy, M.: Qsplat: A multi-resolution point rendering system for large meshes, Proc. SIGGRAPH, p. 343-352 (2000)
13. Samet, H.: Applications of Spatial Data Structures, Addison-Wesley (1990)
14. Subramanian, K.: Adapting search structures to scene characteristics for ray tracing. Ph.D. thesis, Computer Sciences, Univ. Texas at Austin (1990)
15. Teller, S., et al.: Visibility preprocessing for interactive walkthroughs, Computer Graphics, 25(4), 61-69 (1991)
16. Wand, M., Fischer, M., Peter, I., Meyer, F., and Strasser, W.: The randomized z-buffer algorithm: Interactive rendering of highly complex scenes, Proc. SIGGRAPH, p361-370 (2001)
17. Wonka, P., Wimmer, M. and Schmalstieg, D.: Visibility preprocessing with occluder fusion for urban walkthroughs, Rendering Techniques 2000:11th Eurographics workshop on rendering, p. 71-82 (2000)
18. Zhang, H., Manocha, D., Hudson, T., and Hoff, K.: Visibility culling using hierarchical occlusion maps, Proc. SIGGRAPH, p.77-88 (1997)

Image-Based Walkthrough over Internet on Mobile Devices

Yu Lei¹, Zhongding Jiang², Deren Chen¹, and Hujun Bao²

¹ College of Computer Science, Zhejiang University, Hangzhou 310027, P.R. China
lei.yu@zju.edu.cn, dr.chen@zj.edu.cn

² State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China
{zdjiang,bao}@cad.zju.edu.cn

Abstract. Real-time rendering of complex 3D scene on mobile devices is a challenging task. The main reason is that mobile devices have limited computational capabilities and are lack of powerful 3D graphics hardware support. In this paper, we propose an Image-Based Rendering (IBR) system for mobile devices to visualize real-world or synthetic scenes in network environment. Our system uses server for computing the required image segments of pre-captured panoramic video, and transmitting them to client. After receiving data, mobile client carries out rendering using simple image warping. The rendering process needs less computational power and is insensitive to the scene's complexity. A rate-control scheme is designed for efficient use of network bandwidth for handling network congestion. Pre-fetching and cache management are also considered on client and server sides for efficient memory use and reducing transmission request. With this client-server architecture and local rendering scheme, interactive exploration of 3D scene on mobile devices becomes possible. Experimental results show that our system can achieve acceptable rendering speed on common mobile devices.

1 Introduction

The ultimate goals of grid computing will enable us to efficiently utilize the various computing resources on network in a safe manner. In network environment, there are many visualization applications that can benefit from grid computing, such as interactive exploration of 3D scene, tele-presence, virtual tour and online games.

One method to realize these applications is first selective downloading geometry data to client, then renders it using graphics hardware. X3D is one of the candidate technologies[14]. However, this scheme cannot achieve real-time performance on mobile clients. First, the huge data volume of complex 3D scene cannot be loaded into memories of mobile devices. Second, real-time rendering such data set needs powerful 3D Graphics Processing Units (GPUs), and fast Floating Point Units (FPUs). However, they are usually unavailable on mobile clients.

Image-Based Rendering (IBR) is an alternative to traditional geometry rendering[12]. It can synthesize photo-realistic novel views using recorded images

captured from real-world or synthetic scenes. The rendering cost of IBR is independent of scene's complexity. Some IBR methods can be carried out on PC without 3D hardware support. IBR is more suitable for visualizing 3D world on mobile devices since the rendering process requires less computational resources than that of traditional geometry rendering.

Authoring 3D scene using IBR representation requires lots of images. It is impossible for mobile devices to load the entire data set into their memories. Fortunately, a small parts of the whole data set are needed for rendering at one viewpoint. Therefore, only the required parts should be transmitted. Pre-fetching data that will be used in a short of time is important for a practical system. Caches with carefully designed replacement schemes are also important for efficient data management. With pre-fetching and cache management, the performance can be improved since the required data is commonly available in caches. Similar to other network applications, rate-control scheme is necessary for reducing the influence of network latency and makes the interaction between both sides more efficient.

The remainder of the paper is organized as follows. Section 2 describes the related work. The details of our system are presented in Section 3. The experimental results are shown in Section 4. We draw conclusions and point out the future work in Section 5.

2 Related Work

Remote walkthrough systems are not new, several work have been done in the past. Noimark *et al.* [10] designed a server-based walkthrough system that renders virtual environment into video frames and streams them to clients. Using on-chip MPEG-4 video encoder, the rendering engine generates scene frames according to client's input. However, the reported frame rate is only about $2 \sim 3$ fps due to the expensive compression scheme. Chim *et al.* [2] implemented a distributed walkthrough environment based on the on-demand transmission strategy. Clients render virtual scenes by fetching geometry data from server. A multi-resolution caching mechanism was employed for reducing the influence of network latency. Although this scheme is very useful, it belongs to traditional geometry rendering, and is not suitable for mobile devices. Engel *et al.* [4] presented a framework providing remote control of 3D applications based on Open Inventor or Cosmo3D. It transmitted compressed images from server to java-based client. The client sent its events through CORBA requests. For mobile devices, CORBA is expensive. Ma *et al.* [8] developed an end-to-end, low-cost solution for visualizing time-varying volume data rendered on a parallel computer. The system transmitted compressed images to display devices through a wide-area network. However, their work did not address how to tailor streaming techniques for mobile clients.

The systems described above could not achieve acceptable performance, or are not tailored for mobile devices for remote walkthrough. Due to the varieties of screen size and computational capacities of these systems, porting them to

mobile devices needs much effort. For example, since mobile devices are usually connected to wireless network, tedious work is required to modify the underlying communication protocol and redesign image/video compression algorithms to handle the error-prone wireless connection. Our remote rendering system (Section 3) is designed with these issues in mind, and attempts to resolve them.

3 The Walkthrough System

3.1 System Architecture

Our system carefully partitions the rendering task on both server and client sides, and adaptively tunes the balance between them. Figure 1 shows its architecture. The server is responsible for determining and transmitting the required data to client. With this design, one server can handle hundreds of simultaneous requests from clients. After receiving the data, mobile client carries out rendering locally by simple image warping. The details are described in the following subsections.

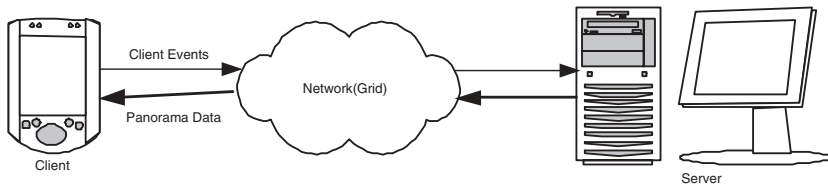


Fig. 1. Overview of the architecture

3.2 The Panoramic Video

There are many image-based representations that can be used to represent a scene. LightField/Lumigraph[7, 6], and Concentric Mosaics (CMs)[11] are well known IBR representations. However, capturing them is not an easy task and usually requires specific devices. Therefore, we choose panorama as rendering primitive of our system. Panorama can be easily acquired by simply rotating one camera mounted on a tripod and capturing a set of images at different directions [1, 9]. After capturing, panorama is produced by carefully stitching the captured images. Rendering panorama is cheap and only uses image warping. Current mobile devices, such as Pocket PC and smart-phone, can render it quickly without 3D graphics hardware support. Panorama can be represented as cubic, spherical, or cylindrical environment map. We choose cubic representation since it can be easily generated using mainstream 3D software. Using simple warping equations, other formulations can be converted to cubic format. Figure 2 shows one cubic panorama illustrating one snapshot of the new campus of Zhejiang University.

For authoring a complex scene, lots of panoramas are needed. We organize these panoramas into panoramic videos (PVs) according to their spatial positions. For representing the whole scene, many PVs are used. One panoramic



Fig. 2. One cubic panorama of the new campus of Zhejiang University

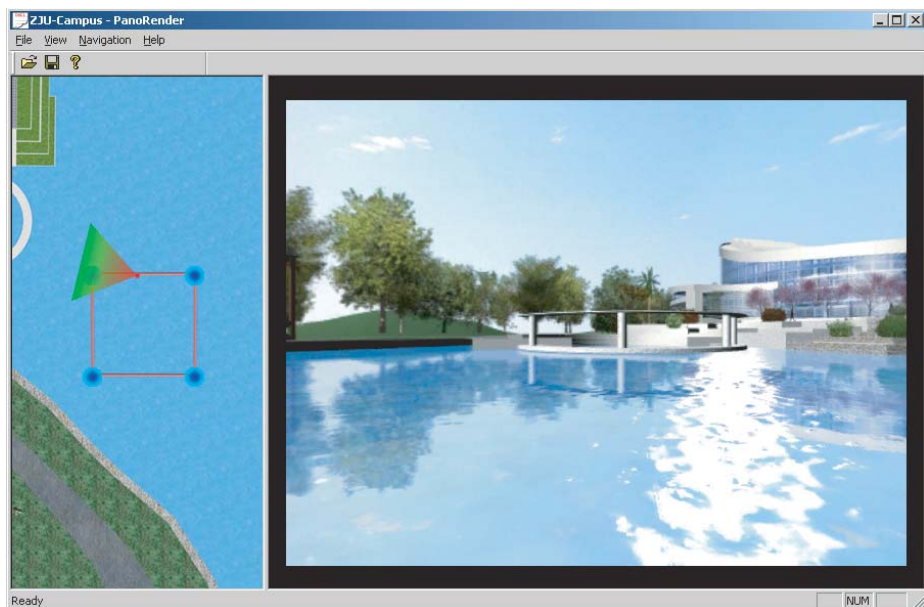


Fig. 3. Panoramic video navigation system on PC

video can be represented as a path in the navigation map (see Figure 3). The panoramic videos can form panoramic video loops. Since each panorama is associated with spatial information, user can move freely along paths in the navigation map. At branching point between paths, the next-navigation-path can be automatically chosen based on user's viewpoint and paths' definitions. Figure 3 shows one snapshot of our panoramic video navigation system running on single PC. The navigation map is shown in the left part. The right part shows the rendering result. The client-server system is built on this system.

The raw data of panoramic videos are huge. It even cannot be totally loaded into mainframe's memory. Therefore, compression algorithms must be employed for reducing data size significantly. Our system adopts JPEG2000 standard [13] to compress panoramic videos. The core algorithms of JPEG2000 are based on discrete wavelet transform (DWT). It supports progressive and region of interest

(ROI) decoding, and error-resilience. These features are very useful for applications on mobile devices with different computational power and screen size, and using wireless network connection. With the same visual quality, JPEG2000 can achieve higher compression ratio than JPEG under low bit-rate conditions. Unlike traditional video coding standards, such as MPEG-1/2/4, H.263/264, we can randomly access individual frame without decoding others by compressing panoramic video using motion JPEG2000. This random access function is necessary for just-in-time rendering in most IBR systems [12].

Since field-of-view (FOV) of the virtual camera is limited, only parts of one panoramic image are required for rendering a novel view. The required parts are also referred as image segments. Therefore, the server can transmit the whole panorama at one position or the required image segments according to user's motion and viewpoint.

3.3 Transmission Scheme

Real-time rendering system should respond to user's input in a short of time. The performance of our system is influenced by network latency and jitter. Networking latency indicates the length of time that incurs when a message gets from end-to-end. Unfortunately, latency cannot be totally eliminated. For interactive 3D applications, network latency ranging from 0.1 to 0.3 second is acceptable. Network jitter indicates the variance of transmission time. It can be compensated by caching several frames before displaying them on client side.

We design rate control scheme to maximize utilization of bandwidth, trying to avoid network congestion for reducing latency. The server probes the network periodically to estimate its bandwidth. Two methods can be used when the bandwidth is not constant. First, the server can dynamically change the data size of image segments according to the remaining bandwidth. This is can be done by progressive transmission the compressed panoramic videos. If the first method does not work well, the server will increase user's step to reduce the number of required image segments to be transmitted.

For transmission of the image segments, we use UDP protocol. Although packet can be lost, it can be quickly delivered over network. We implement the basic idea of real-time-transport (RTP) protocol.

3.4 Cache Management

Caches are used on client and server sides for improving the performance of our system. In server side, since the whole data cannot be entirely loaded into memory, cache is used together with a pre-fetch process to keep I/O operation efficient. The pre-fetching process reads data nearest to user's current position, and the most possible wandering path that user will go through. By swapping data between disk and memory, only a small amount of memory is required.

On mobile client, schemes of cache management are designed for efficient memory use and avoiding retransmission of cached data. We cache image segments in the neighborhood of user's current position. One small circle is used

to specify the neighborhood. The center of the circle is located at user's current position. Its radius is dynamically changed for maintaining a constant memory footprint. The farthest image segments will be discarded when there is no room for the newest received data. We also cache the decoded data for reducing decompression operations. If the client cached a complete panoramic image, it could quickly rotate at that location.

3.5 Interaction Between Client and Server

Touch pen is the most used input device for mobile applications. It is originally designed for word processing software, not for 3D walkthrough. Fortunately, four buttons (UP, LEFT, DOWN, RIGHT) in mobile devices, such as Palms, Pocket PCs and most of smart phones, are useful for 3D navigation. Five wandering manners are defined in our system, namely, Move Forward, Move Backward, Rotate Left, Rotate Right and Stop. We map these buttons to user's wandering manners during navigation process.

Our system works in server-pushing manner. The server maintains a set of parameters describing the client's status. These parameters include user's view-point, wandering manner, and status of caches. The required image segments are determined by them, and sent to the client. If the status of client is changed, server should be notified and update the corresponding parameters.

The change of client's status is mainly due to user's interaction. The interaction events are sent from client to server through a separate network channel. Since interaction event is time critical, UDP protocol is used. For handling packet losing, server will send a packet to acknowledge client's event. If client did not receive the response, it would re-send request packet to server until it is confirmed.

4 Experimental Results

We have implemented our image-based walkthrough system in an IEEE 802.11b wireless network with 11Mbps bandwidth. The server runs on a PC with Intel Pentium IV 2.0GHz CPU, 512 MB memory, and Microsoft Windows 2003 Server Edition. The client runs on a HP iPAQ 5450 Pocket PC with Intel PXA250 CPU, 64MB SDRAM, and Microsoft Pocket PC 2002 operating system with wireless network support. The maximum screen size is 320×240 in pixel resolution. We use 200×200 pixels to display the rendering result.

We captured one region of the new campus of Zhejiang University. 204 panoramic images are captured and each image is 2048×512 . The acquired panoramas are organized into one panoramic video loop shown in the navigation map in Figure 3. The raw data is in 24 bits RGB format, and the total size is 612MB. We compress it using JPEG2000 and JPEG for performance comparison. Figure 4 shows one snapshot of our client-server system.

Table one compares the performance measured in frame-per-second (fps) using different compression standards. As it shown, performance using JPEG compression is better than that of using JPEG2000. The main reason is the decoding

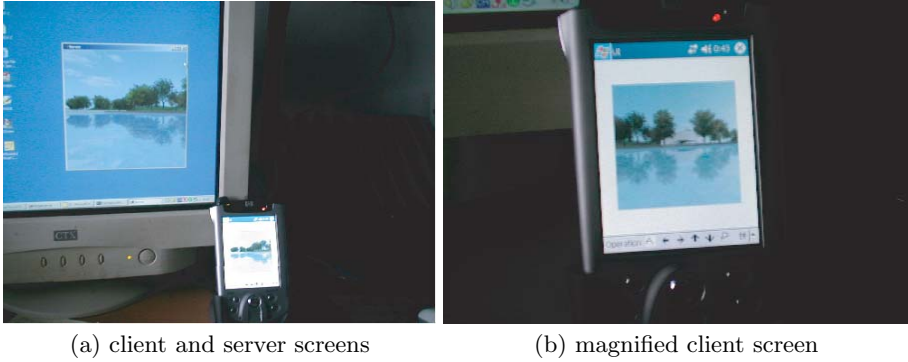


Fig. 4. Snapshot of our system

Table 1. Performance comparison using different compression standards

User Action	JPEG(fps)	JPEG2000(fps)
Rotation	14.3	8.2
Translation	7.8	4.6

computation of JPEG2000 is expensive. In both experiments, rotation action is faster than translation because the decoded data is cached. Although the speed using JPEG2000 can not achieve real-time performance ($>10\text{fps}$), it is still acceptable ($4 \sim 8\text{fps}$) for users using mobile devices.

Our current software implementation of JPEG2000 is unoptimized and not as efficient as product code. Further code optimization can double the speed and achieve real-time performance. Comparing to JPEG, JPEG2000 is more attractive for the near future applications because it is versatile and has many nice features for universal media access, especially for wireless application. Within few years, more mobile devices will support JPEG2000 with hardware, which will significantly improve the performance.

5 Conclusions and Future Work

In this paper, we propose an IBR system for mobile devices to interactively explore 3D scene in network environment. The scene is represented as panoramic videos stored at remote server. Our system works in sever pushing manner. It uses server to compute and transmit the required image segments according to client's status. After receiving data, client carries out rendering locally using CPU and displays the resultant image to its small screen. Our system designs rate control, and cache management schemes for efficient use of network bandwidth and memory resource.

The preliminary implementation of our system uses low speed wireless network connection and PC. Upgrading it to grid computing environment will sig-

nificantly improve its performance as more bandwidth and powerful server are available. Integrating IBR techniques into visualization applications under grid computing has just emerged[3]. We are currently porting our system to grid computing environment equipped with Globus[5]. The grid-version of our system will allow mobile devices to access the image-based scenes stored at server anywhere and anytime.

Acknowledgements

This project is supported in part by Natural Science Foundation of China (Grant No.60021201 and Grant No.2002CB312104) and 863 High-Tech Program of China (Grant No.2002AA414070). The Research Fund for the Doctoral Program of Higher Education (Grant No.20030335083).

References

1. S.E. Chen. QuickTime VR-An Image-based Approach to Virtual Environment Navigation. Proceedings of SIGGRAPH 1995, pp. 29–38, August 1995.
2. J. Chim, R.W.H. Lau. CyberWalk: A Web-Based Distributed Virtual Walkthrough environment. IEEE Transactions on Multimedia, 5(4):503–515, 2003.
3. J. Ding, J. Huang, M. Beck, S. Liu, T. Moore, S. Soltesz. Remote Visualization by Browsing Image Based Databases with Logistical Networking. Proceedings of ACM/IEEE conference on Supercomputing, 2003.
4. K. Engel, O. Sommer, T. Ertl. A Framework for Interactive Hardware Accelerated Remote 3D Visualization. Proceedings of EG/IEEE TCVG Symposium on Visualization 2000, pp.167–177, 2000.
5. <http://www.globus.org/>
6. S.J. Gortler, R. Grzeszczuk, R. Szeliski, M.F. Cohen. The lumigraph. Proceedings of SIGGRAPH 1996, pp.43–54, August 1996.
7. M. Levoy, P. Hanrahan. Light field rendering. Proceedings of SIGGRAPH 1996, pp.31–42, August 1996.
8. K.L. Ma, D.M. Camp. High performance visualization of time-varying volume data over a wide-area network status. Proceedings of ACM/IEEE conference on Supercomputing, 2000.
9. L. McMillan, G. Bishop. Plenoptic modeling: An image-based rendering system. Proceedings of SIGGRAPH 1995, pp.39–46, August 1995.
10. Y. Noimark, D. Cohen-Or. Streaming Scenes to MPEG-4 Video Enabled Devices. IEEE Computer Graphics and Applications, 23(1):58–64, 2003.
11. H.-Y. Shum, L.-W. He. Rendering with concentric mosaics. Proceedings of SIGGRAPH 1999, pp. 299–306, August 1999.
12. H.-Y. Shum, S.B. Kang, S.-C. Chan. Survey of Image-Based Representation and Compression Techniques. IEEE Transactions on Circuits Systems for Video Technology, 13(11):1029–1037.
13. D. Taubman, M. Marcellin. JPEG2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, Boston, 2002.
14. <http://www.web3d.org/>

Service-Oriented RunTime Infrastructure on Grid*

Tae-Dong Lee, Seung-Hun Yoo, and Chang-Sung Jeong**

School of Electrical Engineering in Korea University 1-5ka, Anam-Dong
Sungbuk-Ku, Seoul 136-701, Korea

{lyadlove,friendlyu}@snoopy.korea.ac.kr, csjeong@charlie.korea.ac.kr

Abstract. The HLA (High Level Architecture) is a blueprint to use to develop the necessary infrastructure in order to promote interoperability and reusability within the modeling and simulation community. RTI (RunTime Infrastructure), software implementation of HLA, is composed of three components: libRTI, FedExec, and RTIExec. RTI is a middleware that supports dynamic, many-to-many communication in a distributed environment. Running a large-scale distributed simulation may need a large amount of computing resources at geographically. Such environment raises serious security concerns and dynamic coordination concerns. Motivated by these concerns, we have developed the RTI that can overcome these problems using Open Grid Services Architecture (OGSA) inside Globus Toolkit3(GT3). We call it service-oriented RTI-G. In this paper, we illustrate the structure of the service-oriented RTI on Grid and how it can solve the mentioned problems.

1 Introduction

While High Level Architecture(HLA) [1] is an architecture and is not software based, its core instrument in supporting the runtime services is RTI software. As RTI is an interface specification, it is envisioned that multiple implementations, potentially providing domain specific benefits, will be developed in the future. Running a large-scale distributed simulation may need a large amount of computing resource geographically different locations. HLA provides application developers with a powerful framework for distributed simulation reuse and interoperability, however its design was not intended to support software applications that need to integrate instruments, displays, computational and information resources managed by diverse organizations. Moreover, The existing RTI all do not consider coordinating and managing the resource for distributed simulation to complete the simulation efficiently and effectively. The Grid, however, was originally designed to address precisely those issues. Globus toolkit3 (GT3) [5]

* This work has been supported by KOSEF and KIPA-Information Technology Research Center, University research program by Ministry of Information & Communication, and Brain Korea 21 projects in 2004.

** Corresponding author.

provides several valuable capabilities to RTI. The Grid Resource Broker, Globus Resource Allocation Manager (GRAM), and Metacomputing Directory Service are used to initiate RTI. Previously, the execution of RTI was a painful manual process. With Globus, the execution can be started from a single point. The Globus Monitoring and Discovery Service (MDS) provides a standard mechanism for publishing and discovering resource status and configuration information [7]. Without MDS, we have to foreknow a large amount of resource information at geographically different location. The High Level Architecture (HLA) and its Run-Time Interface (RTI) do also not define support of access controls required to provide necessary protection levels. The HLA do not currently support the authentication for the joining federates. The Globus Toolkit 3(GT3) can make up for the weak points of HLA in the security. Especially, in the military simulation, the security problem is the big issue. The GSI service of GT3 will be the good solution for that. The RTI-G provides security communication between RTI components. To solve above problem, we implemented the service-oriented RTI-G through the combination of the HLA and GT3. This paper describes design and implementation of service-oriented RTI-G for security communications and dynamic environment on Grid. Next, we experiment application on RTI-G and conclude our opinions and discuss the future work [7].

2 Related Works

We categorize the current RTIs into three parts based on developed organization: DoD, Company, University. One of the most popular existing RTIs is RTI1.3NG [2] developed by DoD. The Next Generation Runtime Infrastructure (RTI 1.3NG) was developed using a process that identified the requirements of an RTI, analyzed the key architectural elements, and leveraged the experience gained from previous RTI implementations and other distributed computing systems. The ability to configure and evolve the internal system components was a driving principle for this design. This flexibility was deemed vital to the support of the disparate operating conditions of various federates and federations, as well as to adapt the RTI 1.3NG to future technologies and techniques. RTIs developed by company are pRTI [3] and MAK RTI [4]. In February 2000, pRTI 1.3 became the first commercial RTI to be certified by DMSO. Deploying HLA applications imposes several requirements on an RTI implementation. In addition to speed and stability, availability on different platforms and robustness of the RTI becomes increasingly important. To be able to monitor and debug a deployed system, a simple and self-explanatory graphical user interface should be available. The development of pRTI has been driven by many different needs, several of them seemingly in conflict with each other, flexibility versus ease-of-use, performance versus complexity, etc. The development of the MAK Real-Time RTI primarily came about in response to the difficulties of working with the existing RTI implementations in a development environment. The real-time virtual simulation community concern for RTI performance indicated the need for an RTI that optimized the basic requirements of real-time

simulation. The development of MAK Real-Time RTI focused on the subset of HLA Interface Specification that meets those requirements. The design of MAK Real-Time RTI is based on simplicity and efficiency. At the same time, it does not neglect the use of data abstractions that promote extension and adaptation. It also minimizes the amount of handshaking and synchronization that occurs between RTI components. The RTI developed by university is RTI-Kit which is not fully implemented by Georgia Tech. RTI-Kit is implemented as a modular software package to realize runtime infrastructure (RTI). RTI-Kit software spans a wide variety of computing platforms, ranging from tightly coupled machines such as shared memory multiprocessors and cluster computers to distributed workstations connected via a local area or wide area network.

Regardless of HLA community, distributed computing has been concerned with collaboration, data sharing, and other new modes of interaction that involve distributed resources. The result is an increased focus on the interconnection of systems both within and across enterprises. These evolutionary pressures generate new requirements for distributed application development and deployment. Continuing decentralization and distribution of software, hardware, and human resources make it essential that we achieve desired qualities of service (QoS) on resources assembled dynamically from enterprise systems, service provider systems, and customer systems, which requires the new abstractions and concepts. The solution is OGSA [8]. OGSA allow applications to access and share resources and services across distributed, wide area networks [6].

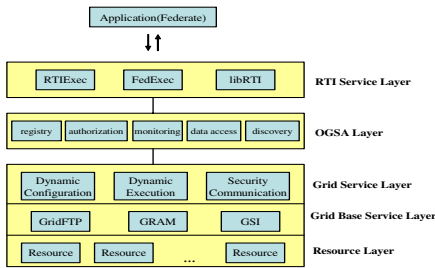


Fig. 1. Layered architecture of service-oriented RTI on Grid

3 Designs of Service-Oriented RTI on Grid

The service-oriented RTI-G is a grid-enabled implementation of the RTI. That is, using services from the GT3, RTI-G allows you to provide dynamic configuration, dynamic execution and security communication. To apply the Grid technologies to RTI, the appropriate design is required. In this section, we discuss these features of the design. Service-oriented RTI-G consists of five layers that are RTI Service layer, OGSA layer, Grid Service layer, Grid Base Service layer and Resource layer.

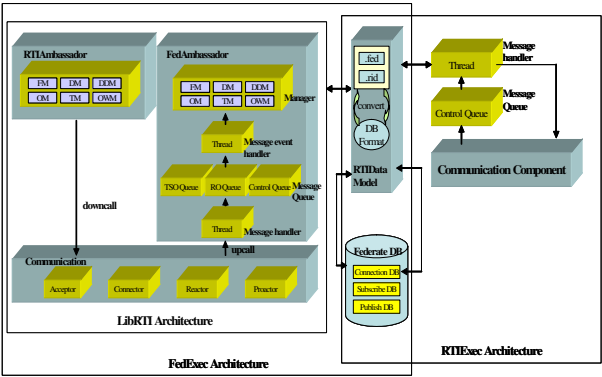


Fig. 2. Overall Architecture of our RTI

3.1 RTI Service Layer

The major components in RTI Service layer are libRTI, FedExec, RTIExec. RTI software can be executed on a standalone workstation or executed over an arbitrarily complex network. The RTIExec process manages the creation and destruction of federation executions. Each executing federation is characterized by a single, global FedExec. The FedExec manages federates joining and resigning the federation. The libRTI library extends RTI services to federate developers. Services are accomplished through encapsulated communications between libRTI, RTIExec, and the appropriate FedExec. The proposed architectures are the multi-layered architectures which can provide a well-defined model of an information system that reflects the scale and depth of the application-level services and separate the application models into discrete tiers such that lower levels have no need for access to services defined at higher levels. The layering provides a way to manage complexity and reuse software and is applicable when a system is divisible into areas of concern with well-defined boundaries. Often, it is undesirable for application developers to know all the details of every software tier in the system, due to complexity, multiple software packages, and platform differences. Layering must provide the architectural boundaries that manage complexity for individual developers.

RTIExec is composed of Communication layer with thread, Control Queue, process layer with thread, Federation DB as shown in figure 2. The communication layer detects and demultiplexes messages and dispatches them to their associated message handler. The message handler encapsulates messages to events and inserts events into control queue. The process layer dispatches events to their associated event handler. The event handler processes events and provides service to requesting components. The RTIExec is a globally known process. Each application communicates with RTIExec to initialize RTI components. The primary purpose of RTIExecs is to manage the creation and destruction of Fed Execs. An RTIExec directs joining federates to the appropriate federation exe-

cution. RTIExec ensures that each FedExec has a unique name. The federation DB has a name DB for managing unique name and address DB for managing connected FedExec.

The FedExec architecture is composed of communication layer, supplier layer, scheduler layer, consumer layer and configuration management layer as shown in figure 2. The components that are supplier layer, scheduler layer, consumer layer decouple method execution from method invocation to enhance concurrency and simplify synchronized access to an object that resides in its own thread of control. The Supplier provides an interface that allows event handlers to invoke publicly accessible methods on an event object using standard, strongly-typed programming language features, rather than passing loosely typed messages between threads. When event handlers invoke a method defined by the Event handle class, Supplier create events and put events into the Scheduler's Activation Queue. The Scheduler runs in a different thread than its supplier layer, managing an Activation Queue. A Scheduler decides which events to dequeue next and execute on the consumer that processes this events. This scheduling decision is based on various criteria, such as ordering. The Consumer defines the behavior and state that is being modeled as an Active Object. Consumer implements the methods defined in the consumer event handle class. A Consumer event handler is invoked when its corresponding event is gotten by a Scheduler. The Configuration management layer initializes components and manages FedExec data that organizes data into Connection, Subscribe, and Publish data. The information about clients connected is stored in Connection data and published and subscribed information each in Publish data and Subscribe data.

libRTI architecture provides the RTI services specified in the HLA Interface Specification to federate developers. The major components in libRTI are summarized as follows: A federate interfaces to the RTI via the RTIAmbassador and FedAmbassador, which present the language specific API to the user. Internally the RTIAmbassador and FedAmbassador convert the supported APIs into a common format before passing service requests and data to other RTI components.

3.2 OGSA Layer

The Open Grid Services Architecture (OGSA) integrates key Grid technologies with Web services mechanisms to create a distributed system framework based around the Open Grid Services Infrastructure (OGSI). A Grid service instance is a service that conforms to a set of conventions (expressed as WSDL interfaces, extensions, and behaviors) for such purposes as lifetime management, discovery of characteristics, notification, and so forth. Grid services provide for the controlled management of the distributed and often long-lived state that is commonly required in sophisticated distributed applications.

Grid Services have the potential to bring remote and decentralized RTI service discovery and invocation to RTI-G from GT3 container. OGSA supports dynamic discovery and separation of the actual protocols from the abstract RTI functionality description. We design and implement the use of Grid core ser-

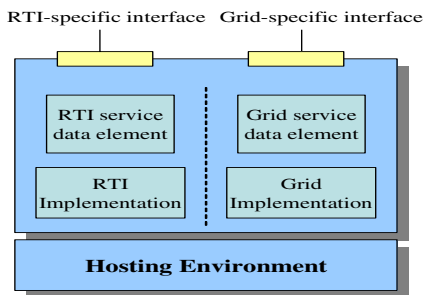


Fig. 3. OGSA division: RTI-specific part, Grid-specific part

vices(GridFTP, GRAM, GSI) for the migration and transport of actual Federate data and RTI components(RTIExec, FedExec, libRTI). We design the OGSA into two parts: RTI-specific part, Grid-specific part like shown in figure 3.

The benefits of component technologies enable encapsulation, modular construction of applications and software reuse. RTI-G components are defined in Open Grid Service Architecture (OGSA) and Infrastructure (OGSI) for the Grid. Using an approach where RTI-G components are modeled as a set of Grid services, which allows for RTI-G components to be compatible with the OGSI specification. This enables RTI-G components to be accessible via standard Grid clients, especially the ones that are portal-based.

3.3 Grid Services Layer

The Grid Service layer provides security communication and dynamic coordination to RTI Service layer, which uses GSI for security communication, GridFTP and GASS for transfer of data. The Grid Service layer has components that are composed of a dynamic configuration and execution and security components. The service that allocates suitable resource for executing FedExec and executes FedExec in remote locations is provided to RTI by a dynamic configuration and execution. The security components provide secured and authenticated communication to RTI. Dynamic configuration and execution consists of three components which are Resource Broker(RB), Simple Transfer Agent(STA) and Remote Execution(RE). The RB informs applications of grid resources, which builds on MDS of Globus Toolkit, leveraging existing functionalities but providing a powerful interface to applications. When user will be launching an application, users know the available and appropriate resources to utilize within the grid. This task could be carried out by a broker function. The RB consists of two parts: one part, resource broker, provides a powerful interface to application user, the other part, GRAP (Grid Resource Allocation Policy) is a user-defined policy. The GRAP decides priority order of information that get through MDS. The STA provides speed and reliability for files being transferred. These files can be executables, scripts, or other modules representing the programs that will be run remotely,

job dependencies, for example dynamic shared libraries and results files. Globus Toolkit uses the GridFTP protocol for all file transfers. File transfer is built on top of a client/server architecture that implies that a GridFTP server must be running on the remote node to be able to transfer a file to the remote host. The globus-io module and Globus GASS subsystem transparently uses the GridFTP protocol. When a job is submitted by a client, the request that job is executed is sent to the remote host. The RE is responsible for the execution, which builds on GRAM of Globus Toolkit and provides a user-friendly interface to application.

4 Experiments

The implementation of service-oriented RTI-G is based on Linux using C++ program language. The scenario of experiment is as follows. The object information is described in figure 4(c), and specification of computers in this experiment is shown figure 4(d). The first experiment is that four clients connect to Server1. After connection, RtiExec is created and makes FedExec generated as Grid service through GT3 container. The second experiment is that four clients connect to Server1 like first experiment. After connection, RtiExec service is created and makes FedExec service generated through fork operation from GT3 container. However, FedExec service is generated on best server (Server2 in this experiment) by RB, STA, and RE service in RTI-G. The hub of communication is FedExec whose performance is decided on resource states. After generation of FedExec, 100 objects are tested, and 100 objects are increased per 50 seconds. We measure the data transfer rate on computer executing FedExec. The result is shown figure 4(e). The performance of RTI-G is better than that of RTI.

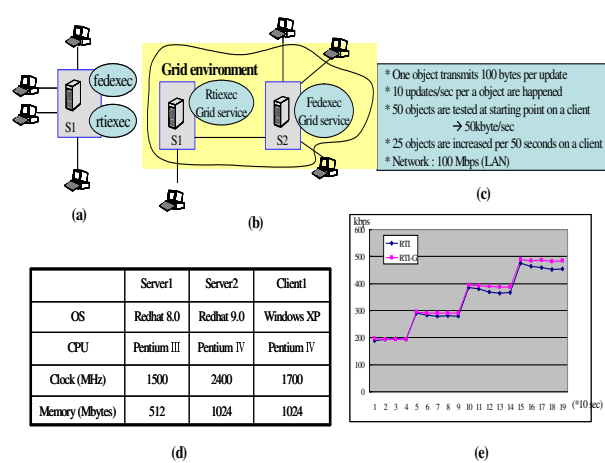


Fig. 4. Experiment (a)RTI experiment (b) RTI-G experiment (c) Test condition (d) Server/Client specification (e) Experiment result

5 Conclusions

RTI (RunTime Infrastructure), software implementation of HLA, is composed of three components: libRTI, FedExec, and RTIExec. RTI is a middleware that supports dynamic, many-to-many communication in a distributed environment. Running a large-scale distributed simulation may need a large amount of computing resources geographically. Such environment raises serious security concerns and dynamic coordination concerns. Motivated by these concerns, we have developed the RTI that can overcome these problems using Globus Toolkit3(GT3). We called it service-oriented RTI-G. In this paper, we have illustrated the structure of the service-oriented RTI on Grid and how it can solve the mentioned problems. The first step in this paper has introduced the overview of HLA and OGSA which includes the formulation of a conceptual framework, the specification of the data model, the interface, and the semantics of the event service. The following step, we have designed the basic architecture to implement service-oriented RTI-G composed of RTI service layer, OGSA layer, Grid service layer, Grid base service layer and resource layer. The last part of this paper, we have shown how to implement and experiment service-oriented RTI-G.

References

1. IEEE Standard for Modeling and Simulation (M&S), "High Level Architecture (HLA) Federate Interface Specification," IEEE Std 1516.1-2000
2. T.B. Stephen, J.R. Noseworthy, J.H. Frank, "Implementation of the Next Generation RTI," 1999 Spring Simulation Interoperability Workshop.
3. M. Karlsson, L. Olsson, "pRTI 1516 - Rationale and Design," 2001 Fall Simulation Interoperability Workshop.
4. D.W. Douglas, "Rationale and Design of MAK Real-Time RTI," 2001 Spring Simulation Interoperability Workshop.
5. I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," Intl J. Supercomputer Applications, 11(2):115-128, 1997.
6. I. Foster, C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," June 22, 2002.
7. T.Jianquan, T.Weiqing, D.Jingbo, C.Lizhi, "MOM-G: message-oriented middleware on grid environment based on OGSA," Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on , 20-23 Oct. 2003 Pages:424 - 427
8. D.Talia, "The Open Grid Services Architecture: where the grid meets the Web," Internet Computing, IEEE , Volume: 6 , Issue: 6 , Nov.-Dec. 2002 Pages:67 - 71

Quality-of-Service Driven Visual Scheduling in Grid Computing

Changqin Huang^{1,2}, Guanghua Song^{1,2}, and Yao Zheng^{1,2}

¹ College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China

² Center for Engineering and Scientific Computation, Zhejiang University
Hangzhou, 310027, P.R. China

Abstract. To make full use of grid resources and to meet users' requirements, efficient scheduling is a key concern in grid environments. Aiming at grid-based engineering computation applications, this paper proposes a Quality of Service (QoS) driven user-centric scheduling strategy. Firstly, degree of credit and degree of guarantee are defined, and aggregate utility ratio is modeled as a composite QoS; Secondly, for different types of grid users, two scheduling methods and steering-enabled visual interfaces are presented, respectively; Thirdly, four performance metrics and aggregate utility ratio are visualized to facilitate the user's interaction with scheduling; Finally, corresponding post-scheduling mechanisms are designed to cope with scenarios where scheduled tasks could not obtain expected QoS. This study is part of a grid project, MASSIVE, and the experiments show that the visual scheduling strategy presented is suitable for computational grids.

1 Introduction

Grid computing is becoming a new computing infrastructure for scientific computing and cooperative works, and it promotes users' collaboration through flexible and coordinated sharing of distributed resources. The performance that a grid can deliver varies dynamically due to resources competing, network status, task type, and so on. In order to improve the performance of a grid, it is necessary to provide applicable mechanisms that can perform effective task scheduling in the grid. Our investigation on existing grid scheduling methodology indicates the following two aspects: (1) Although many performance metrics are concerned, such as the system utilization, throughput, turnaround time, and waiting time, aggregate metrics are seldom considered, meanwhile, QoS of scheduling is only considered insufficiently; (2) Among a variety of scheduling methodologies, in general, scheduling mechanism is only regarded as a part of underlying infrastructure. That is, they are oriented to grid systems, not to grid users, and they don't provide users with capabilities of more convenient steering.

To overcome the weakness of the conventional grid scheduling, this study concentrates on user-centric grid scheduling to better satisfy users' requirements. Our scheduling approach models an aggregate utility ratio as a composite performance metric,

i.e. QoS, and the grid user's requirements can be met by improving four performance metrics and the aggregate utility ratio. As the use of visualization is beneficial for understanding and analyzing computational information, visualization is utilized in our user-centric scheduling, to enable necessary interactions conveniently happen between users and the system. Furthermore, unexpected results of scheduling are remedied based on the thresholds of QoS during post-scheduling.

This paper is organized as follows. The next section introduces the related works. Section 3 presents a composite user-concerned QoS model in scheduling, and gives definitions of four performance metrics. Visual scheduling framework is described in Section 4. While the automatic and manual visual scheduling interfaces, and QoS based visual steering are addressed in Section 5. Section 6 studies a post-scheduling mechanism with concern of QoS, and the last section outlines the conclusions and future work.

2 Related Works

To meet system and users' requirements in grid environments, a variety of scheduling strategies and algorithms are proposed. Buyya et al. [2] propose a scheduling algorithm with only concern of two performance metrics: cost and time. Cheng et al. [3] study the feasibility problem of scheduling a set of start time dependent tasks with deadlines and identical initial processing time, however, they set strict constraints (e.g. a single machine). Beaumont et al. [4] aim at the scheduling of independent, equal-sized tasks and improve the performance by making full use of a system metric (bandwidth). Based on time-varying resource prices, Dogan et al. [5] consider the problem of statically scheduling a set of independent tasks with multiple QoS requirements. He et al. [6] introduce the matching of the QoS request and service between the tasks and hosts based on the conventional Min-Min algorithm. However, the QoS is only concerned with the completion time, and scheduling is made between the two differentiated types: the high QoS tasks and low QoS tasks. Chen et al. [7] incorporate QoS management into Open Grid Services Architecture (OGSA) and provide a high-level middleware to build complex applications with QoS guarantees. The job scheduling is oriented to the service grid, and the QoS focuses on Success Ratio and In-Time Ratio. Abeni et al. [8] introduce a statistical guarantee of deadline based on inter-arrival and execution time probability distributions. However, it is more applicable to real-time system, than to grid environments. Chun et al. [9] present a scheduling approach based on resource markets and focusing on user-centric performance. Between the above studies and ours, there exist a difference: the formers aim at non-interactive scheduling in clusters, but not visual scheduling like ours. Islam et al. [10] provide QoS with the response time given by the end user in the form of guarantees of the completion time for submitted independent parallel jobs, however, they haven't considered aggregate QoS and visual steering yet.

Utilizing visualization is a good thought in grids. Shalf et al. [1] investigate the numerous issues of implementing grid-enabled distributed visualization, and advise a distributed visualization architecture. Whereas, visual steering of scheduling is not

their emphasis. Jiang et al. [11] propose a rule-based visualization mechanism for a computational steering collaboration, allow users to extract regions of interests to visualize, and track and quantify the evolution of these features in grid environments. Our work gives a visual scheduling control and visual performance presentation. Bonnassieux et al. [12] concentrate on automated resource, service discovery and monitoring, and design a flexible grid visualization tool to represent all corresponding virtual views needed. However, scheduling and QoS are not considered in [11,12].

3 Composite Quality of Service (QoS) Model in Scheduling

In practice, the popularization of grid applications relies greatly on grid user's concerns, therefore user-oriented QoS is very important. At present, budget/cost and deadline/time have been introduced as parameters of QoS. Grid resources are normally highly dynamic and heterogeneous, whilst the tasks to be scheduled dynamically arrive for execution across Vos. Thereby, more performance parameters should be applied to reflect the actual characteristics of grids. Here, in addition to budget/cost and deadline/time, we introduce two metrics: degree of credit and degree of guarantee. Moreover, we define aggregate utility ratio as composite QoS, where the scheduler will make a dynamic schedule. The composite QoS model is shown in Figure 1.

A user oriented QoS, in the form of aggregate utility ratio, is composed of four performance parameters with respective weights during composition. All these performance metrics affect the scheduling by information change with the scheduler, and all required values corresponding to a certain task can be inputted via a graphical interface. After a user's task is scheduled, all values will be displayed to give users for a reference, and serve for the post-scheduling if necessary. The definitions of all performance metrics are given as follows.

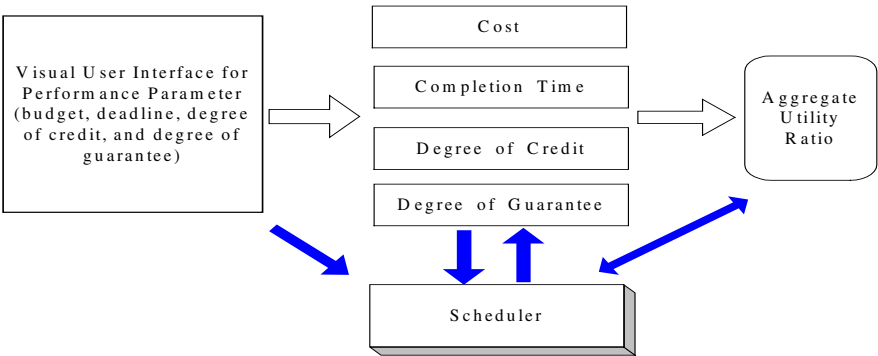


Fig. 1. A diagram of quality of service model and interaction relations.

Definition 1. Cost C is the amount of “money” based on a pay-in-use mechanism. Assume N denotes the number of used resource units. Let UT denote the used time of used resources for the task, and P associated “price” of one unit of used resource in

one unit of time. Under the conditions of a uniform “money” unit, for a task of a certain user, cost C is defined as

$$C = N * UT * P. \quad (1)$$

Definition 2. Completion time CT is defined as the wall-clock time at which nodes complete a certain task (after having finished any previously assigned tasks)[6]. Let AT denote the arrival time of the task, ST the starting time of the task, and ET the expected execution time. From the above definitions, we have

$$CT = ST + ET. \quad (2)$$

Definition 3. Degree of credit DC denotes the success ratio of the actual service provided by resources across VOs. In this study, DC is only used for the entity of grid nodes, and it is gained by computing the historical information in activity profiles of nodes. Let TA denote the number of tasks once accepted, and TC the number of tasks completed under the constraints of users. Then DC is defined as

$$DC = TC/TA. \quad (3)$$

Definition 4. Degree of guarantee DG denotes the probability of task completion before the deadline. Let ET denote the expected execution time, ST the starting time of the task, D the deadline of this task, and P the accurate ratio of predicted execution time. Then DG is defined as

$$DG = (P * P^{(D-ST)/ET}) / (1-P), \text{ if } D-ST \geq ET; \text{ otherwise } DG = 0. \quad (4)$$

In the above definitions, there are two concepts to denote the uncertainty of a grid: degree of credit DC and degree of guarantee DG . The expected execution time ET is a prediction value (produced by performance predictor), and the used time of a certain resource UT is a pre-scheduled time with respect to the pair of this resource and an associated task by the scheduler.

Let B denote the user budget, then we can define the composite quality of service (Aggregate Utility Ratio, AUR) as follows

$$AUR = \lambda_1 (B/C) * \lambda_2 (D/CT) * \lambda_3 DC * \lambda_4 DG \quad (5)$$

In Eq. (5), $\lambda_1, \lambda_2, \lambda_3$ and λ_4 stand for the weights of the four performance factors, and they can be set by the administrator or specified by the VOs.

4 Visual Scheduling Framework

To improve the quality of scheduling and to provide users better control of steering, the scheduling could be performed in a visual manner, through two types of visual windows, scheduling and QoS sessions. These visual methods provide users with a direct awareness and a friendly interaction. A visual scheduling framework is shown in Figure 2, with the following features.

1. According to different types of users, the scheduling is performed in both manual and automatic manner with a simple monitoring mechanism.
2. In addition to the arrived task queue, a reserved task queue is used for arranging pre-scheduled tasks. Hence reservation-based scheduling can perform well.
3. The scheduling algorithms during the automatic scheduling can be selected by users, as various conventional algorithms can be integrated into the system. Moreover, there is further extensibility for inclusion of new algorithms. The idea is based on the fact that different algorithms are suitable for respective environments and objectives.
4. Performance predictor serves for the scheduler by predicting and computing the previously mentioned performance metrics. It analyzes the performances of the tasks to be scheduled in advance, furthermore, it evaluates all QoS parameters of scheduled tasks. The values are provided to users in a visual fashion.
5. Quality of service manager is responsible for accepting these required performance values from the input, for setting performance threshold values that are conditions of triggering adjustment mechanisms and warning user, and for managing the events of post-scheduling and the interactions for a better performance.

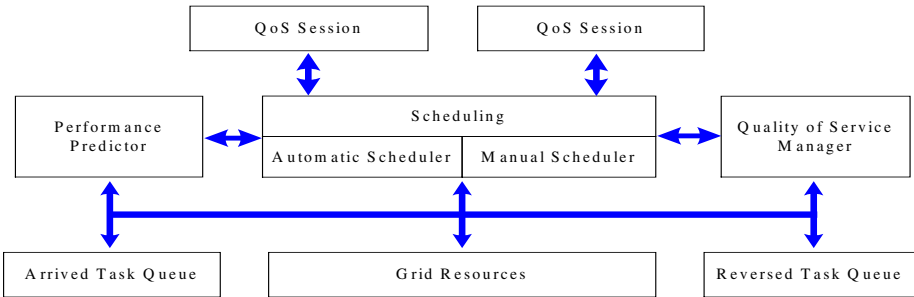


Fig. 2. A visual scheduling framework.

5 Implementation of Visual Scheduling

We have developed a visual grid prototype system oriented to engineering computing, named MASSIVE (formerly VGrid [13]). This study is a part of the MASSIVE project and we adopt Globus Tools 2.4 as an underlying middleware and a development tool on Linux systems. The QoS model and the visual scheduling framework are implemented with the aid of the KDevelop package. All visual sessions are refreshed every certain interval or are triggered by associated events.

Figures 3 and 4 show a manual scheduling session and an automatic scheduling session, respectively, where the following details are noticeable.

1. Both of the two visual scheduling sessions give an area, where the results of monitoring are displayed, and there are three operations: watch “RSL file”, “Reschedule” and “Submit to run”, by which interactions with users can happen.

- Tasks are scheduled under all constraints including diverse performance requirements, and performance predictor aids the scheduler to make decisions. In our study, a simple prediction module oriented to engineering computing is developed to serve that purpose.
- In Figure 4, the right middle area is designed for steering these tasks in “Reserved Task Queue”.

Visual steering for quality of service is shown in Figure 5, of which the following details are remarkable.

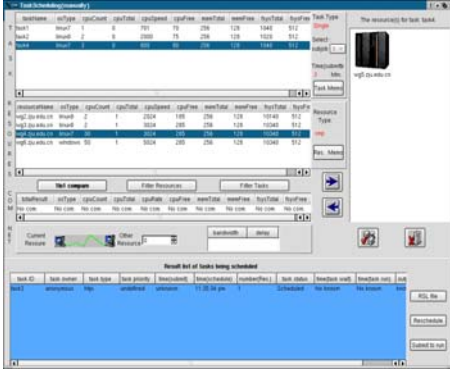


Fig. 3. A session of manual scheduling.



Fig. 4. A session of automatic scheduling.

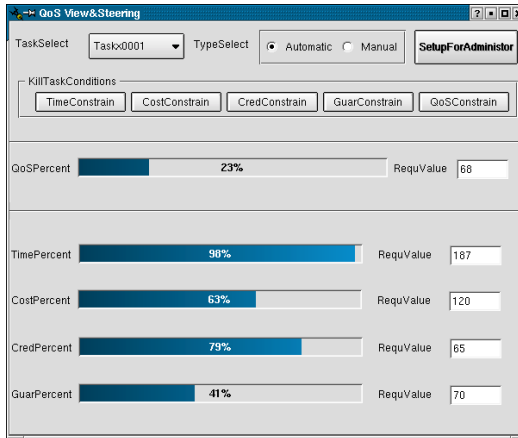


Fig. 5. Visual steering for quality of service.

- The button “SetupForAdministrator” is designed for some important operations to QoS management. For instance, all threshold values of performance metrics can be set via this entry point.
- All values of four types of performance are presented in the form of percentage, and 100% denotes that it is best value in the viewpoint of the submitting user. Similarly, quality of service, as “QoSPercent”, indicates composite performance for the current selected task.

3. Grid users can set the initial performance value in the corresponding “RequValue” area. In the row for “Time”, the input value is used as its deadline, and the input value is used as its expected “monetary” budget in the row for “Cost”. That is, the above two inputs are set with the corresponding actual values, and the rests are percentages.
4. Grid users can also set the initial value in the corresponding “KillTaskConditions” area to decide whether to cancel their scheduled task when a certain performance does not meet the given requirements.

6 Mechanism of Post-scheduling

If there exist some troubles during the scheduled tasks’ execution, perhaps the above mentioned performances do not satisfy users’ requirements anymore. Thereby, the robust scheduling requires an excellent mechanism of post-scheduling. In this study, our basic scenario is: These running tasks will be progressively stopped when the values given by the predictor reach the threshold values or exceed the initial constraint values. Under the guidance of user-centric thought, the operations of post-scheduling can be conducted in either manual manner or automatic manner. If users have set the corresponding “KillTaskConditions”, the system will firstly check these conditions, and if it matches one of them, the scheduled task will be killed, and associated node profiles are modified to affect the future performance prediction. Except for this previous case, post-scheduling performs one of the following actions, according to the set rules and the specified conditions.

1. Kill the task, release the current resource(s) and modify the corresponding profiles.
2. Let the task continue running on the current node(s), meanwhile, let this task run on one or many new nodes in parallel. If someone among these nodes or sets of nodes completes execution of the task, then the rests will cancel their tasks and release themselves. Modify the corresponding profiles.
3. Kill the task and release the current resource(s), meanwhile, let this task run on one or many new nodes in parallel. If someone among these nodes or sets of nodes completes execution of the task, then the rests will cancel their tasks and release themselves. Modify the corresponding profiles.
4. Kill the task and release the current resource(s), after that, put this task into “Arrived Task Queue” or “Reserved Task Queue” to let the scheduler reschedule. Modify the corresponding profiles.
5. Save the necessary information and migrate the task to one or more new nodes, release the old resource(s), and then let these new nodes perform in parallel. Lastly, modify the corresponding profiles.

At present, we only implement the former four actions in the manual manner by coupling with the above scheduler. More issues of design and implementation of the post-scheduling will be studied in our future work.

7 Conclusions and Future Work

The user-oriented Quality of Service (QoS) is a key to popularizing grid applications. However, no integrated solution has been well addressed to meet users' requirements during grid scheduling. In this paper, we have modeled a new composite quality of service and its associated performance metrics, such as degree of credit, and degree of guarantee, which progressively reflect the grid quality status. Aiming at the requirements of engineering computation applications, a QoS driven visual scheduling framework is proposed. For different types of grid users, two scheduling methodologies and steering-enabled visual scheduling interfaces are designed and implemented, respectively. Four performance metrics and an aggregate utility ratio improve users' capability of steering QoS-driven scheduling in a visual fashion. Finally, corresponding post-scheduling mechanisms are designed to cope with cases where scheduled tasks could not obtain expected QoS. We have conducted some experiments in a test bed named MASSIVE. They show that this visual scheduling approach is suitable for computational grids.

In the future, we plan to study the technologies of performance prediction in grid computing in the area of scientific and engineering computation, and to use further cases to test this visual scheduling prototype. Also, we are ready to study the automation of post-scheduling, migration of tasks, and recovery mechanisms in depth.

References

1. J. Shalf and E. W. Bethel: The Grid and Future Visualization System Architectures, *IEEE Computer Graphics and Applications*, 23(2), 6-9, 2003.
2. R. Buyya, et al.: A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids, *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 2002.
3. T. Cheng, et al.: Scheduling Start Time Dependent Tasks with Deadlines and Identical Initial Processing Times on a Single Machine, *Computers & Operations Research*, 30: 51-62, 2003.
4. O. Beaumont, et al.: Bandwidth-Centric Allocation of Independent Tasks on Heterogeneous Platforms. *Proc. of the International Parallel and Distributed Processing Symposium*, 2002
5. A. Dogan and F. Özgüner: Scheduling Independent Tasks with QoS Requirements in Grid Computing with Time-Varying Resource Prices. *Proc. of Grid Computing-GRID 2002*, 2002.
6. X.S. He, X.H. Sun, and G.V. Laszewski: QoS Guided Min-Min Heuristic for Grid Task Scheduling, *Journal of Computer Science & Technology*, 18(4): 442-451, 2003.
7. H.H. Chen, H. Jin, et al.: Early Experience in QoS-Based Service Grid Architecture, *AP-Web 2004, LNCS 3007*, pp. 924-927, 2004.
8. L. Abeni and G. Buttazzo: QoS Guarantee Using Probabilistic Deadlines, *Proc. of the IEEE Euromicro Conference on Real-Time Systems*, 1999.
9. B.N. Chun and D.E. Culler: User-centric Performance Analysis of Market-based Cluster Batch Schedulers, *Proc. of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

10. M. Islam, et al.: QoS Based Scheme for Parallel Job Scheduling, *JSSPP 2003*, LNCS 2862, pp. 252-268, 2003.
11. L. Jiang, H. Liu, et al.: Rule-Based Visualization in a Computational Steering Collaboratory, *ICCS 2004*, LNCS 3038, pp. 58-65, 2004.
12. F. Bonnassieux, et al.: Automatic Services Discovery, Monitoring and Visualization of Grid Environments: The MapCenter Approach, *Across Grids 2003*, LNCS 2970, pp. 222-229, 2004.
13. G. Wei, Y. Zheng, et al.: An Engineering Computation Oriented Visual Grid Framework, *GCC 2003*, LNCS 3032, pp.51-58, 2004.

AFEC: An Advanced FEC Algorithm for Video Transmission Control over the Grid*

Yubo Tan, Yuxing Peng, Sikun Li, and Fujie Chen

PDL, School of Computer, National University of Defense Technology
Changsha, Hunan, 410073, P.R. China
tanyubo@sina.com, BenenTan@msn.com

Abstract. Real-time video communication is very important to the Internet applications of video conferencing, video telephony, video-on-demand, and etc. However, the heavy traffic of video data along with its timing constraints makes it a challenge to provide large-scale, high QoS media streaming service over the current Grid environment. This paper presents a novel video transmission control algorithm AFEC (Advanced FEC), which is based on the FEC coding technology and the KALMAN filter theory. By modifying the rate using the adapted KALMAN filter, this algorithm efficiently solves the problem of rate fluctuation caused by the loss of the ACK packets. It also weakens the influence of the loss of elementary layer packets (or other important contents) in transmission and provides the continuity of the video transmission over Internet. Simulation results indicate that this algorithm can guarantee satisfied performance for video transmission in networks of high packet loss rates.

1 Introduction

With the development of the network technology, the grid-based video applications grow rapidly. Real-time video communication over grid is attracting a lot of attention to the applications, such as video conference, video telephony, and video-on-demand, etc. Due to its sensitivity to network delay and packet loss ratio, video transmission is usually based on unreliable transport protocols, like UDP. To provide satisfied QoS for video applications under available network capacity becomes a crucial issue.

The theory of layered multicast ^[1, 2, 3] can provide multi-level video quality over the different networks. According to the theory, video stream is encoded to different quality layers. The server sends each video layer over a separate multicast group. A receiver periodically joins a higher layer's group to explore the available bandwidth. If packet loss is detected after the join-experiment, the receiver will leave the group. This control loop continues during the transmission. The layered theory is considered as a promising approach for adaptive video transmission. First, it is fully compatible with the current best effort Internet infrastructure. Second, it is scalable and works well with heterogeneous receivers because adaptation is performed by the receivers.

* This work is supported by National 973 Foundation of China under Grant No. 202CB312105.

Current surveys [8, 15, 16] on layered transmission are based on the assumption that the elementary layer has been received correctly. However, due to the congestion of the Internet, this assumption cannot always be guaranteed. To solve this problem, W. Tan et al. [5] proposed two more FEC coding layers under elementary layer and they used ERC (Equation-based Rate Control) to adjust the rate. But their algorithm hasn't considered the delay of FEC coding. Moreover the algorithm uses a low-pass filter, which is not sensitive to the instantaneous bandwidth. Lee et al. [4] described the packet-level and byte-level FEC coding which are used in wire-less communication. But no effective algorithm was given. R. Puri et al. [7] proposed LIMD/H (History) control algorithm based on LIMD (Linear Increase Multiplicative Decrease), but their algorithm doesn't guarantee the continuity of video transmission.

In order to eliminate the influence of the loss of the elementary layer packets and congestion during video transmission, we adapt the traditional FEC [9, 10, 11] coding method and propose a Kalman^[6] filter based transmission control algorithm AFEC(Advanced FEC), which only encodes the elementary-layer packets and some important data using the packet-level FEC method, and greatly reduces the quantity of the ack-packets. Algorithm avoids the traffic congestion on network and the oscillatory of the sending rate, efficiently decreases the delay of video transmission and guarantees the continuity of video transmission. Experiments show that our algorithm is efficient and it can provide better video service in the network of high packet loss rates.

2 Transmission Control

The layered transmission technology [12,13,14] can satisfy the requirement of different clients with different bandwidth. But if the loss ratio of the network is too high to guarantee the elementary layer, it is hard to provide satisfied QoS video service for clients only using layered transmission technology. In order to provide large-scale and high QoS video service in grid environment, in which heavy video traffic always leads to high loss ratio; we propose a novel AFEC algorithm, which is based on FEC and Kalman theory, to provide better services under this condition. The overall framework of the AFEC algorithm is given in Fig. 1.

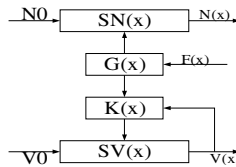


Fig. 1. The framework of AFEC

In Fig. 1 $V(x)$ (sender rate) and $N(x)$ (the quantity of sending packets) are output variables we want. The constants $N0$ and $V0$ are the initial values. $F(x)$ stands for the actions of ACK packets. $K(x)$ is the Kalman Filter. $G(x)$ processes the information of

ACK packet, which provide the control of the output and the information of timeout. $SN(x)$ deals with the quantity of sending packets and $SV(x)$ is the sender rate processing.

The algorithm includes the following parts:

2.1 Coding and the State-Model

The sender encodes m original packets and gets $m+k$ FEC-packet (systematic coding), reference to Fig. 2 The control mechanism includes two parameters, i.e., the current sender rate $V_{current}$ and the quantity of sending packets b , which can avoid the network congestion and keep the integrity of packets to provide continuous video service.

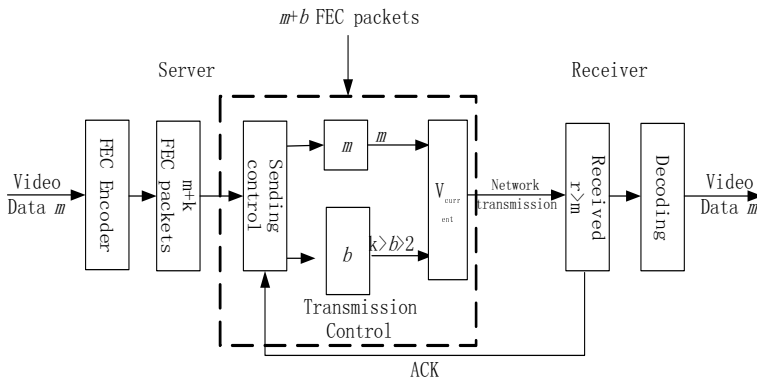


Fig. 2. Encoder and Decoder

The server sends $m+b$ FEC-packets, and the variable b depends on the current network bandwidth ($k > b > 2$). If the status of bandwidth is worse, the variable b tends to k , otherwise to 2. The receiver can restore the m original video packets only if the sum of received packets r is greater than m [9].

2.2 The Information of ACK and Estimating the Network Bandwidth

The server calculates the rate V through RTT (Round Trip Time) and the loss ratio p , and sends the result V back to the server. The current TCP stable sender rate can be calculated by these parameters [20]: the size of TCP packet s , the packet loss ratio l , the timeout t_0 , and RTT time t_{RTT} .

The receiver does not send the ACK packet for each received packet, while a set of packets. The ACK packets include the status about received packets.

2.3 Modifying the Sending Rate

The server calculates the sender rate with the received ACK information by KALMAN filter, and sets the current sender rate V_{current} . If the ACK packet loses, the server set the current sender rate V_{current} by the predicted value of KALMAN filter. Let $\underline{V}_k$ be the rate got from ACK packets, V_k be the predicted value by KALMAN filter, $\overline{V}_k$ be the calculated value by KALMAN filter, $\overline{P}_k$, P_k be the calculate factor, V_{current} be the current sender rate, and A, B, R, Q be constants.

Initialization:

$$V_0 \leftarrow V_{\text{init}}; P_0 \leftarrow 1; V_{\text{current}} \leftarrow V_0 \quad (1)$$

The server recursively calculates the predicted result as follows:

$$\overline{V}_k = A * V_{k-1} + B \quad (2)$$

$$\overline{P}_k = A * P_k + Q \quad (3)$$

$$V_{\text{current}} = \overline{V}_k \quad (4)$$

According to the rate from ACK packet, the server modifies these parameters immediately as follows:

$$K_k = \frac{P_k}{P_k + R} \quad (5)$$

$$V_k = \overline{V}_k + K_k * (\underline{V}_k - \overline{V}_k) \quad (6)$$

$$P_k = (1 - K_k) * \overline{P}_k \quad (7)$$

If the ACK packet is lost ,the server sets the sender rate as follow:

$$V_{\text{current}} = V_k \quad (8)$$

2.4 Control the Quantity of Sending Packets

As a result, we get the state model, based on Gilbert model^[14], about the quantity of FEC packets, which need to be sent as follows (Figure 3).

The status 0 means that the variable b is fixed and the server sends packets normally. The status 1 means that the receiver successfully receives packets for s times, then the server decreases the value of variable b . The state 2 means that the receiver losses packet for t times, then the server increases the value of b . The parameter p is the probability of successfully received packet, while q is the probability of loss.

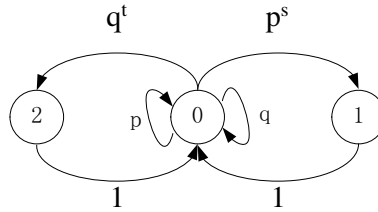


Fig. 3. AFEC state model ($t>1, s>1$)

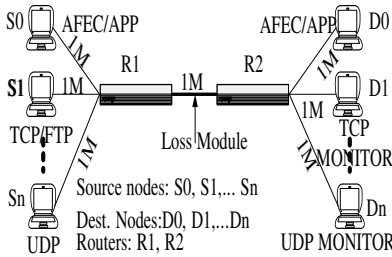


Fig. 4. Topology

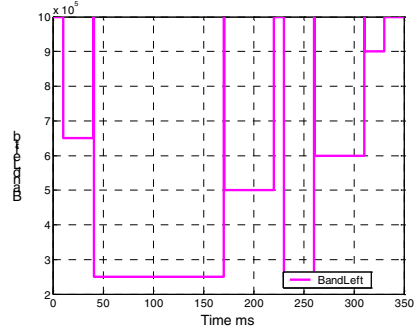


Fig. 5. The variation of bandwidth between R1 and R2

3 Performance Evaluation

We use NS2 [12, 13] to simulate the performance of the AFEC algorithm. The network topology is given in figure 4.

Experiment 1: We use (12, 6) FEC to encode the packets ($m=6$), and the server sends $m+b$ ($2 \leq b \leq 6$) packets. We select a fixed clip, total 754 I-frames, and compare the sum of received valid elementary packets. Figure 5 shows the available bandwidth between R1 and R2 according to the time. Under sending fixed clip, we compare the performance of AFEC and NOAFEC. Figure 6 is the comparison of sender rate. Figure 7 is the received valid packets of elementary layer. From figure 6, we can see the curve of AFEC is changed gently and the amplitude is smaller. The server sends packets placidly. We calculate the received elementary layer packets showed in figure 7. If up to 6 different packets are received, the original packets can be decoded successfully, otherwise retransmission wanted.

If the sum of received packets is less than 6, the receiver cannot decode the video data correctly. During the period, AFEC algorithm only failed to decode video 4 times while the normal transmission failed up to 16 times. Under the same network condition, AFEC algorithm can provide better service.

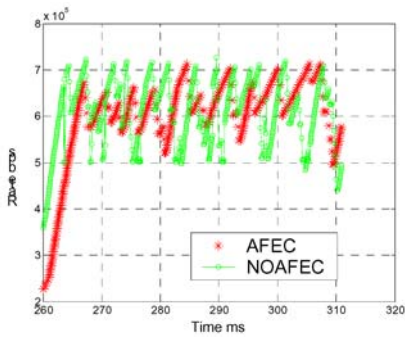


Fig. 6. The micro view of the sender rate of AFEC and NOAFEC while sending the continuous packets

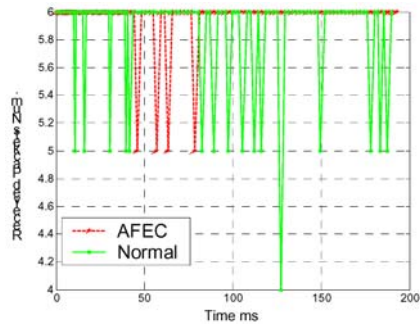


Fig. 7. The received packets of elementary layer under fixed clip

Experiment 2: We compare the sum of received elementary layer packets when the loss ratio changed . We simulate the loss ratio of grid network by changing the variable p of the Loss Module in NS2. The bandwidth between R1 and R2 is given in figure 5. With the loss ratio higher, the performance of the AFEC algorithm is more remarkable. The PSNR^[18] ratio is given in figure 8.

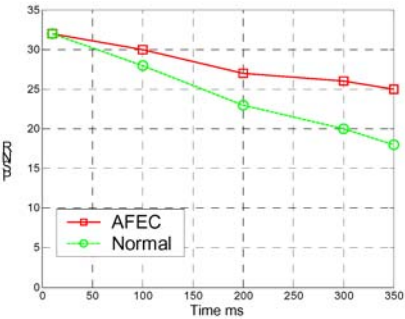


Fig. 8. The PSNR ratio

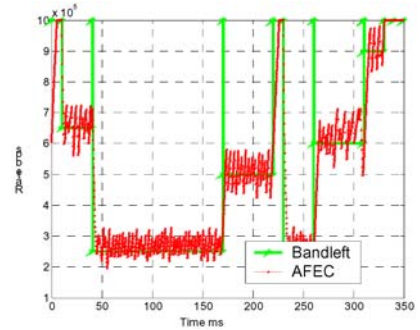


Fig. 9. The sensitivity of AFEC to the bandwidth

4 The Performance of AFEC Algorithm

The adaptive ability of the AFEC algorithm is given in figure 9. From the figure, we can see that the AFEC algorithm is sensitive to the bandwidth. When the bandwidth is changed, the server quickly adapts with the AFEC algorithm. Figure 10 shows the curve of the AFEC algorithm under different loss ratio. The parameters p is set to 0.002, 0.01, 0.02, 0.05 and 0.1. Respectively with the loss ratio higher, the ability of occurring bandwidth becomes less sensitive. Under different loss ratio, we shows the times of received all the valid packets (figure 11). With the loss ratio higher, the re-

ceived time is more postponed. That means in the network of high packet loss rates, the receiver or the server should keep enough buffer to sustain the algorithm.

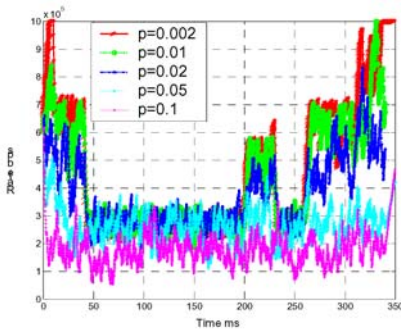


Fig. 10. The rate of AFEC under different loss ratio

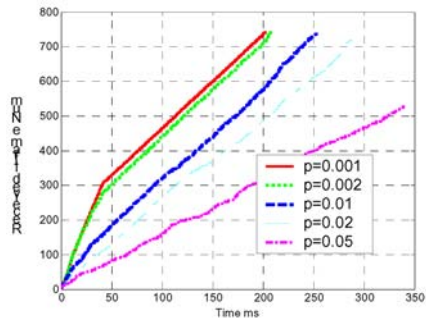


Fig. 11. The circumstance of received packets about the elementary layer under the different loss ratio

5 Conclusion

We propose a novel AFEC algorithm to support video transmission in grid environment. The algorithm modifies the sender rate with the adapted KALMAN filter and the quantity of FEC encoded packets, it can avoid the collapse caused by the mass ACK packets and the rate fluctuation caused by the loss of the ACK packets. Algorithm can guarantee video transmission quality even in the network of high packet loss rates, which maybe leads to loss of the elementary layer. This algorithm is also suitable for the wireless and P-to-P transmission.

References

1. Turetti, S. Parisi, and J. Bolot, "Experiments with a layered transmission scheme over the internet," Tech. Rep., Technical report RR-3296, INRIA, France, Nov. 1997.
2. Steven McCanne Van Jacobson Martin Vetterli "Receiver-driven Layered Multicast" ACM SIGCOMM '96" On the Use of On-Demand Layer Addition (ODL) with Multi-layer Multicast Transmission Techniques"
3. Lorenzo Vicisano, Jon Crowcroft, and Luigi Rizzo, "Tcp-like congestion control for layered multicast data transfer," in Proc. of IEEE INFOCOM, March 1998, vol. 3, pp. 996 – 1003.
4. T.-W. A Lee, G. Chan, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Optimal Allocation of Packet-Level and Byte-level FEC in Video Multicasting over Wired and Wireless Networks", IEEE Globecom'01, Nov. 2001.
5. W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression" IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 3, pp. 373–87, Mar. 2001.
6. Greg Welch and Gary Bishop "An Introduction to the Kalman Filter"
<http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>

7. R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "Application of FEC-based Multiple Description Coding to Internet Video Streaming and Multicast," Proc. Packet Video Workshop, Forte Village Resort, Sardinia, Italy, May 2000
8. Christoph Neumann, Vincent ROCA "Scalable Video Streaming over ALC:a solution for the Large Scale Multicast Distribution of Videos" INRIA Research Report Quantity 4769, March 13th, 2003
9. "Forward Error Correction Encoding and Decoding" Intel Corporation Technical Staff May 1990
10. Injong Rhee "Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet" ACM SIGCOMM '98 PROGRAM
11. L. Rizzo "Dumynet and Forward Error Correction" In Proceedings of zreenix '98. June 1998
12. McCanne and S. Floyd. ns Network Simulator. <http://www.isi.edu/nsnam/ns/>. and the manual: Kevin Fall, Kannan Varadhan, and the VINT project. The ns manual.
13. Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in Network Simulation. IEEE Computer_, V. 33(N. 5), pp. 59-67, May, 2000.
14. M. Yajnik, S. Moon, and J. Kurose et al., "Measurement and modeling of the temporal dependence in packet loss," UMASS CMPSCI, 98-78, 1998.
15. John Byers, Michael Frumin, Gavin Horn, Michael Luby, Michael Mitzenmacher, Alex Roetter, and William Shaver, "Flid-dl: Congestion control for layered multicast," To appear in the Second Int'l Workshop on Networked Group Communication (NGC 2000), Nov. 2000.
16. Pascal Frossard Olivier Verscheure "Joint Source /FEC Rate Selection for Quality-Optimal MPEG-2 Video Delivery" IEEE 2001 p1815-p1825
17. M.W.Garrett and W.Willinger, "Analysis, modelling ,and generation of self-fimilar VBR video traffic" preprint, 1994.
18. Zhang Yongjun, Zhangyi, Pengyuxing, "Error Control and Recovery Technology in Real-time Media Stream Transmission" Journal of National University of Defense Technology, 3rd volume 2003 p75-76

Delayed State Consistency in Distributed Virtual Environments*

Yuxing Peng, Yongjun Zhang, and Sikun Li

School of Computer, National University of Defense Technology
Changsha, Hunan P.R. China
peng_Kevin@etang.com

Abstract. One of the challenging issues in Distributed Virtual Environments is the consistency maintenance among the participants, or entities. In this paper, we focus on exploring state inconsistency of the entities caused by the network transport delays. Firstly, we enumerate all kinds of the possible state inconsistency instances and the causes. By analyzing the causes, we can conclude that the key to maintain the delayed state consistency is to keep a consistent arriving sequence of the events occurred in the environments. Based on the view, a uniform sequence process scheme and a distributed virtual environment model are proposed respectively to make the view practical for the distributed simulation applications. Finally, our distributed virtual naval battle environments based on the model are given to test if the view is effective. Our experimental study shows that our view can be successfully used for the applications in distributed virtual environments.

1 Introduction

The term “Distributed Virtual Environments” was proposed in the mid 1990s to refer to the distributed computer and network systems, which are applied to simulations, especially to the large-scale battlefield simulations^[1,2].

One of the challenging issues in the distributed virtual environments is the delayed consistency problem^[3]. The problem can be simply depicted as the inconsistency among the entities in the simulation processes due to the network transport delays in the environments. For example, the computer N_1 simulates the entity airstrip and the computer N_2 simulates the entity airplane respectively. If the airstrip was ruined at time t_1 and the result or the state of the airstrip arrived at N_2 at time t_2 , then N_2 did not know the airstrip has been ruined during the time period t_1 to t_2 and the airplane can take off at this time, which causes the state inconsistency between the airstrip and airplane in the point of view of the environments.

This problem prevails in the field of simulation, therefore it draws researchers' attention either from theory or practice's aspect. In recent years, consistency maintenance in distributed virtual environments has been studied by many researchers. The schemes that have been proposed can be summarized as follows^[4-8]: timestamp, local lag

* This work is supported by the Chinese 973 program. The program number is 2002CB312105.

algorithm, time warp, dead reckoning and so on. In addition to the schemes, other mechanisms, such as Lock and Token, are also introduced in distributed virtual environments. These techniques, though helpful, have their limitations depending on the context.

So, in this paper, we try to go a step further into this issue. We focus on the consistency maintenance of entity states, which means that the inconsistent states of the entities will be prevented from occurring in the simulation process. The contribution of the paper is that we use uniform sequences instead of time such timestamp to maintain the consistency maintenance of entity states.

2 Causes of the Inconsistency

What are the causes of the inconsistency? They can be enumerated as follows:

Cause 1: The unexpected arriving sequence of independent events

E_1 and E_2 are two events occurring in the computer N_1 and computer N_2 respectively. The two events can change the state of entity O. Normally, E_1 occurs before E_2 . Because of the network transferring delays, E_2 reaches to entity O before E_1 probably.

There are two cases involved in this situation. One case is that E_1 and E_2 are two independent events. In this case, different arriving sequence results in different simulation results. As shown in figure 1, entity O is a car stopping at position A and facing north. E_1 is a command “go ahead 5 blocks” and E_2 is a command “turn right and go ahead 3 blocks”. If the arriving sequence is E_1 before E_2 , the final position of O is B. If the arriving sequence is E_2 before E_1 , the final position of O is C.

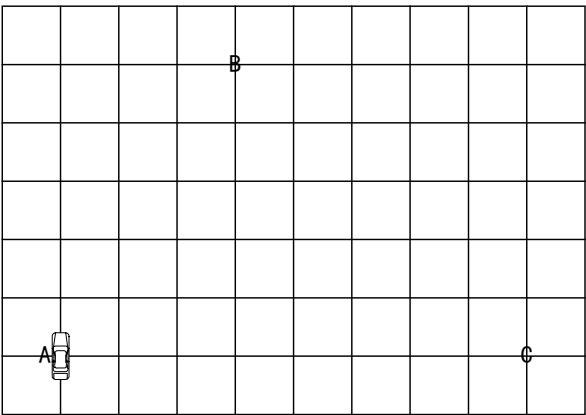


Fig. 1. Different simulation results

Another case is that E_1 and E_2 are two dependent events. In this case, the unexpected arriving sequence results in the inconsistent simulation results. Given the example introduced in introduction, entity O is the airplane and E_1 is the event about the

airstrip has been ruined and E_2 is the event about the airplane's taking off. E_2 is an event occurred naturally in the computer N_2 but it is an unacceptable event, which shouldn't occur in the first place, from the computer N_1 's perspective.

Cause 2: The inconsistent network transferring delay of an event

Given a scenario, E is an event occurred in the computer N at time t . The event is transferred to the computer N_1 and N_2 . Because of the different network transferring delays, E arrives at N_1 at time t_1 and at N_2 at time t_2 . This cause does not result in the inconsistency directly. But if an event occurs during t_1 to t_2 , some inconsistent results are very likely to be generated. Back to the example introduced in introduction, if there are two airplanes being simulated in two different computers and the two airplanes take off at a time period t_1 to t_2 , then the simulation results are one of the airplane took off and another panged.

Cause 3: Other causes

Except for cause 1 and cause 2, there exist other causes, which lead to the inconsistency. But the causes are the combinations of cause 1 and cause 2. For example, more than two events reach to a entity, or a event is transferred to more than one computers, or more than two events reach to more than one entities located in different computers.

3 Delayed State Consistency of the Entities

How can the inconsistent states of the entities be processed not to exist in the distributed virtual environment? Let's analyze the examples given above. In the example illustrated in figure 1, we cannot achieve the final position of entity O in advance. But where the final position is not important to us sometimes. The more important thing is that the arriving sequence is consistent to all the computers involved in the simulation process. If they all consider the arriving sequence is E_1 before E_2 , B is the final position. If they all consider it the other way around, C would be the final position. In the example introduced in introduction, if all the computers consider that the time E_1 arrives at N_2 as the real time E_1 occurs in the distributed virtual environments, not the time E_1 occurs in N_1 , then E_2 cannot result in the inconsistent states of airplane O , because the airplane takes off before the airstrip being ruined. In the example introduced in cause 2, if we let the two airplanes take off after they two know the airstrip being ruined, the inconsistent states will be prevented.

The analysis supports us to believe that it is practicable to maintain the state of the entities consistency. The rational behind our belief lies in: for cause 1, the time when the events occur or arrive is not important. The important thing is the arriving sequence of the events, which should be identical with all the entities in the environments. For cause 2, how much difference the network transferring delays are is not important. The important thing is let event E arrive at all the related computers before any other events arrive, which also means identical arriving sequence of the events. Based on the consideration, the problem of state inconsistency of the entities caused by cause 1 and cause 2 can be solved by giving an identical arriving sequence of all the events.

Because cause 3 is the expansion of cause 1 and cause 2, if there exists an identical arriving sequence for all the events, any inconsistent states of the entities caused by cause 3 can be also prevented.

4 A Uniform Sequence Process Scheme

To guarantee a consistent arriving sequence, we present a uniform sequence process scheme. The scheme can be described as follows. Each entity is simulated in its local computer. The events and the related states, which were initiated from the entities, are sent in to an event process server immediately. The event process server is the server that accepts all the events and the states, and then sends them to the all the computers, which are concerned about the events, including back to the original computer where events occurred. The time of the occurrence of the events was measured as when the server recognized it. So each event has a unique timestamp and all the computers see it in a consistent view. The state is considered to be the current state of the entity in all the involved computers in this step and can be used as the initial state of the next simulation step.

How does an event transfer in the distributed virtual environments and affect the system when no uniform sequence process scheme is used? Given entity O is an entity in computer N_0 and a change of the state of the entity O as event E . E needs to be sent to k computers N_i , $i=1, 2, \dots, k$ and the network transferring delays sending E from N_0 to N_i are t_i , $i=1, 2, \dots, k$. If E occurs in N_0 at time 0, the maximum network transferring delay is $\max\{t_i | i=1, 2, \dots, k\}$ and the time between N_0 knows E and the last computer knows E is also $\max\{t_i | i=1, 2, \dots, k\}$. The maximum network delay reflects the real-time characteristic of the environments and the time between the first and the last reflects the time-space consistency.

How does an event move in the distributed virtual environments when the uniform sequence process scheme is used? Suppose that the event E arrives at the server from computer N_0 at first and the time when the event leaves from the server is t , and the time when E arrives at computer N_i from the server is t_i , $i=0, 1, 2, \dots, k$. Then the time when the first computer knew E is $t+\min\{t_i | i=0, 1, \dots, k\}$ and the time when the last computer known E is $t+\max\{t_i | i=0, 1, \dots, k\}$. The time between the first computer and the last computer is $\max\{t_i | i=0, 1, \dots, k\} - \min\{t_i | i=0, 1, \dots, k\}$, $t \leq \max\{t_i | i=0, 1, \dots, k\}$.

5 A Distributed Virtual Environment Model

We presented a model of the distributed virtual environments based on the scheme we proposed above. As shown in Figure 2, the model consists of the entities, the event processing server and the communication medium. Through the communication medium, the states of the entities arrive at the server and then the server dispatched them to all the related computers where the entities live. The states from the server can be used as the current states of the entities.

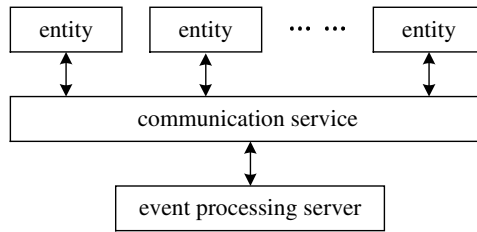


Fig. 2. The architecture of distributed virtual environments model

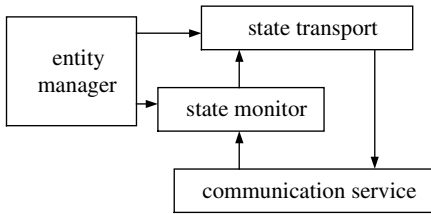


Fig. 3. The event-processing server

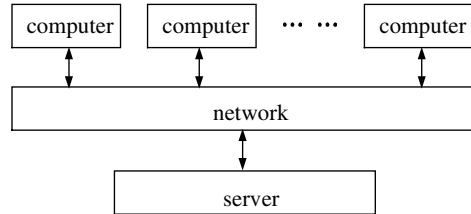


Fig. 4. The architecture of the environments

The server consists of the entity manager, the state monitor and the state transport, as shown in figure 3. The entity manager manages the static information of the entities. The state monitor accepts the dynamic information of the entities from the computers in real time. The state transport sends the states to the related computers where the entities live.

The architecture of our distributed virtual environments based on the model consists of the computers, the server and the network, as shown in figure 4. The computers simulate the residing entities and send their states to the server through the network. More than one entity can reside in a computer and an entity can live across many computers. The server accepts the states of the entities and then sends them to the related computers through the network. Upon receiving the dispatched events from the server, the computer continues simulating the next states of the entities residing.

6 System and Experiment Results

DVSE2000^[9] is our distributed virtual naval battle environments. As an independent system, it can be used in the naval battlefield simulation. Combined with DVNET^[10] developed by Beijing University of Aeronautics and Astronautics, it can be used in the land forces, navy and air force consolidated battlefield simulation. The architecture of DVSE2000 is shown in figure 5.

The uniform sequence process scheme is experimented on DVSE2000. We test the many examples included in cause 1, and cause 2, and cause 3. When the scheme is not used, the example can cause the inconsistency. When the scheme is used, the states of the entities can be maintained consistent during the experiment. But compared with the environments where no uniform sequence process scheme is used, the scheme results in a worse responsiveness in real-time, because it demands the states must

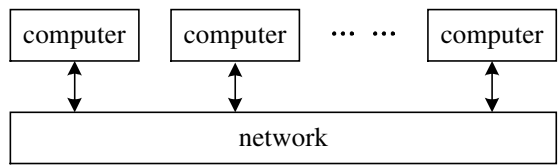


Fig. 5. The architecture of DVSE2000

being transferred by the server. So experimenting on real-time responsiveness is the main target of our experiment.

The problem of real-time responsiveness is caused by states transfer delays when the uniform sequence process scheme is used. We test the delays when the states transfer from sending computers to receiving computers along the path as shown in figure 6. The states are sent to the server by the sending computers and then are sent to the receiving computers by the server. The network bandwidth is 1.5Mbps. Each data package of the state is 2.4Kb. Table 1 shows delays(Sec.) of 20 data packages in a row. The experiment results show that the real-time responsiveness can meet the need of our battlefield simulation applications in distributed virtual environments and the simulation moving image results shown on the screens of the computer can also meet the need of the simulation operators.

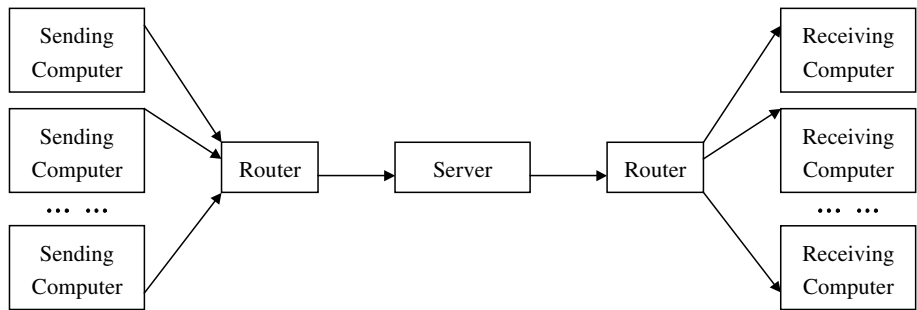


Fig. 6. The states transferring path

Table 1. The experiment results

0.290481456	0.327391348	0.293317839	0.336040612	0.337050865
0.325282016	0.351087021	0.359669786	0.324501686	0.340665993
0.321105949	0.286434808	0.352854621	0.345916962	0.328278136
0.335234976	0.275256322	0.274064987	0.349644673	0.328006053

References

1. E. Frécon and M. Stenius. Dive: A scaleable network architecture for distributed virtual environments. Distributed Systems Engineering Journal, Vol. 5, No. 3, 1998, pp. 91 - 100.
2. R.M. Fujimoto. Parallel and Distributed Simulation Systems, Wiley Interscience, January 2000.

3. M. Mauave. Consistency in replicated continuous interactive media. In Proceeding of ACM Conference on Computer Supported Cooperative Work, Philadelphia, PA, USA, pp. 181-190, December 1998.
4. C.A.Ellis and S.J.Gibbs. Concurrency Control in Groupware System, Proceedings ACM SIGMOD International Conference on Management of Data, Portland, Oregon, June 1989, pages 399-407.
5. Gole and K.Morris. Dead-Reckoning for Aircraft in Distributed Interactive Simulation Proceedings of ALAA Flight Simulation Technology Conference. SC, USA, 1992
6. C.Sun and D.Chen. Consistency Maintenance in Real-Time Collaborative Graphics Editing systems, ACM Transactions on Computer-Human Interaction, Vol.9, No.1, pp.1-41, March 2002
7. B.Roehl. Distributed Virtual Reality – An Overview,
<http://www.ece.uwaterloo.ca/broehl/distrib.html>
8. S.Zhou and et al. Time-Space Consistency in Large Scale Distributed Virtual Environment,
<http://www.ntu.edu.sg/home/asspzhou/ts.ps>
9. L.B.Yan, S.K.Li, State-Oriented Local Lag in Distributed Battlefield. Journal of System Simulation, 2003.9
10. X.P.Chao. DVENET Distributed Environment, Science and Technology Publishing Company, China, 2002

Visual Semantic Query Construction in Dart Database Grid*

Yuxin Mao, Zhaohui Wu, and Huajun Chen

Grid Computing Lab, College of Computer Science, Zhejiang University
Hangzhou 310027, China
{maoyx,wzh,huaajunsir}@zju.edu.cn

Abstract. This paper proposes an effective mechanism to visually construct Semantic Query based on Semantic Browser for query processing in Dart Database Grid. With our approach, distributed database resources are dynamically mapped to a mediated Web Ontology and we can build a universal and intuitive conceptual view against the ontology. Complex query construction is reduced to a set of direct and convenient interactions in the visual semantic view of Semantic Browser and end-users can gain useful information from Database Grid by performing visual Semantic Query operations.

1 Introduction

The development of the Web has resulted in a great deal of distributed and heterogeneous database resources and the structure of different databases can be totally different though they may describe the same thing, so accordingly there are various database querying clients in different applications. The Grid [1] technology is expected to have the ability to resolve the problem of large-scale information resources sharing in dynamic, multi-institutional virtual organizations, which deliver a number of Grid Services [2]. Dart-Grid [3] is an OGSA [4]-based Database Grid developed by Grid Computing Lab of Zhejiang University, which is intended to integrate disparate database resources as a virtual organization in an open, dynamic and wide-area environment. In Database Grid like Dart-Grid, the manners of organizing information may vary in heterogeneous databases, so it's necessary to develop a universal Grid client for end-users to query useful information from distributed database resources, different from traditional database clients developed towards applications separately. As part of the Dart-Grid project, Semantic Browser [5] is such an intelligent client to Database Grid, which manipulates large-scale database resources at semantic layer and provides end-users with a universal and intuitive view for visually querying disparate database resources. Semantic Browser is a lightweight client and interacts with Dart-Grid through different kinds of Grid Services to provide users with a series of semantic-based interactions.

* The work is supported by China 973 fundamental research and development project: The research on application of semantic grid on the knowledge sharing and service of Traditional Chinese Medicine; Intel / University Sponsored Research Program: DartGrid: Building an Information Grid for Traditional Chinese Medicine; and China 211 core project: Network-based Intelligence and Graphics.

2 Visualization of Semantic Information

2.1 Semantic Information

Although schemas are different, we can still extract similar semantics from databases used in the same field. RDF(S) [6] is a kind of simple lightweight ontology representation language and we can use RDF(S) to integrate database resources into virtual organizations. In data-intensive field like TCM [7], we need to introduce semantic information for integrating disparate databases and enable large-scale information sharing in Grid environment. We classify various TCM concepts into 8 top classes of an ontology and each can be subdivided into sub-classes. Based on the shared ontology, we can dynamically create Semantic Mapping between semantic information and distributed databases through the Semantic Register Service of Dart-Grid. The RDF model is directly connected with the schema of relational databases [8].

Definition 1. The following items define a generic semantic mapping:

- (1) $M_{ci}(Table_1, Table_2 \dots, Table_n) = Class_i$;
- (2) $M_{pij}(Field_{i1}, Field_{i2} \dots, Field_{in}) = property_j$;
- (3) $M_i = \langle M_{ci}, M_{pi1}, M_{pi2} \dots, M_{pin} \rangle$; (M_i is a semantic mapping.)
- (4) If M_i is a semantic mapping, then a record in $Table_i$ can be mapped to a direct instance belonging to $Class_i$.

By creating semantic mapping through Semantic Registration, database resources can dynamically join virtual organizations and we can then construct Semantic Query in an intuitive and universal semantic view.

2.2 Semantic Visualization

The Grid Services of Dart-Grid will not directly return original data records in databases to the client, in contrast, semantic information is transferred to Semantic Browser. The semantic information in Dart-Grid owns three characteristics:

- **Complex:** Unlike HTML contents, which are readable for human being, semantic information aims at machine processing. The structure of semantic information is unsuitable for end-users to read directly, so it's very necessary to represent semantic information in an intuitive manner.
- **Large-scale:** Dart-Grid is a large virtual organization with many database resources for large-scale application and the semantic information from Grid Services is also large-scale in most cases.
- **Multiform:** The serviceData about a Grid Service instance is XML-like message and the shared ontology is in RDF(S), which will be updated to OWL [9] soon.

To give end-users an intuitive and universal view on various semantic information from Grid Services, Semantic Browser provide an mechanism to visualize semantic information as intuitive relational graph, which is defined as *Semantic Graph*.

Definition 2. A generic semantic graph is defined by the following items: (1) an acyclic relational graph with a central concept or instance of semantic information is a semantic graph; (2) nodes in a semantic graph are labeled with a semantic link and a set of <arc, node> pairing with a set of inter-operations constitute a semantic graph; (3) two joint parallel semantic graphs with no cross also constitute a semantic graph.

If we want to display semantic graphs clear without loss of information, we should think much of layout and appearance factors. When the scale of semantic information returned from Grid Services is terrifically huge, the structure of corresponding semantic graph gets so complex that a lot of nodes and arcs will overlap with each other in limited user area. To resolve the problem, Semantic Browser slices semantic information according to the granularity of semantics and adopts the radial layout algorithm [10] to arrange the global layout of a semantic graph by each slice of semantic information to avoid overlapping (seen figure 3).

In order to deal with the multiformity of semantic information and get a better effect of visualization, we develop an XML-based visual graph language, Semantic Graph Language (SGL) [5]

```

SGL ::= '<SGL>' namespaceList, graph* '</SGL>'
graph ::= '<graph' idAttr depthAttr '>' subgraph* '</graph>'
subgraph ::= '<subgraph' idAttr typeAttr weightAttr '>' root,
(arc, node | subgraph)* '</subgraph>'
root ::= '<root' idAttr resourceAttr localnameAttr labelAttr
radiusAttr? SonAttr? AngleAttr? SpaceAttr? DisplayAttr? '>'
'</root>'
arc ::= '<arc' idAttr resourceAttr directionAttr '>' '</arc>'
node ::= '<node' idAttr ((resourceAttr localnameAttr labelAttr
| (literalAttr operatorAttr inputAttr)) angleAttr? spaceAttr?
DisplayAttr? '>' '</node>'

```

to visualize various semantic information as semantic graphs. SGL takes semantics into account and treats them as part of graph elements. The SGL BNF definition can be referenced in my previous papers and above there are fragments of the definition. A graph can be divided into

hierarchy sub-graphs and each sub-graph represents a concept or instance as well as its <property, property value> pairs. Two important attributes of sub-graph:

- **Type:** in sub-graph, there are four types of basic semantic relations named as *class-class*, *class-instance*, *instance-property*, and *correlative*. Semantic Graph gives different apparent parameters to different types.
- **Weight:** weight is an important factor in the radial layout algorithm. The weight value of a sub-graph directly decides its proportion in the graph room.

Definition 3. The weight of a sub-graph can be calculated by the equation:

$$weight = \alpha \cdot n_s + \beta \cdot n_l \quad (1)$$

Here n_s is the sub-graph number of the sub-graph and n_l is the leaf number of the sub-graph. A sub-graph is mainly composed by a root, edges with nodes and can be nested. Semantic Browser has the built-in support for converting various formats of semantic information into SGL stream and this form of SGL document is very direct to be processed.

3 Semantic Query Construction

3.1 Typical Working Process

In Semantic Browser, each user operation will acquire semantic information from the Ontology Service of Dart-Grid through a URI [11]. Semantic Browser drives SG-Factory to construct semantic graphs according to the semantic information fed back. After Semantic Registration, Semantic Query can be constructed visually in the semantic view during the process of Semantic Browse [12]. The Semantic Query request is exactly processed by the Semantic Query Service, dispatched among the nodes of the virtual organization and transformed into local SQL query by an engine. Query results are returned from the virtual organization as semantic information, which will be processed and visualized as semantic graphs (see figure 3).

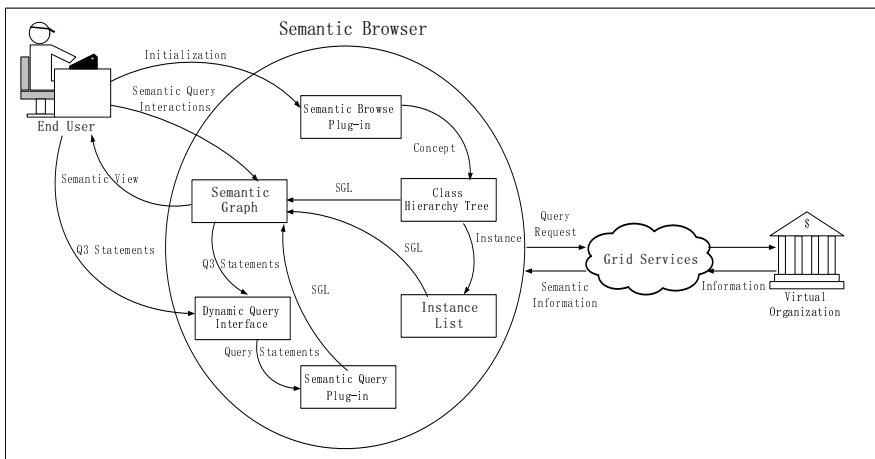


Fig. 1. A typical process of Semantic Query construction in Semantic Browser

With Semantic Browser, users interact with virtual organization through Grid Services at semantic layer rather than querying in local databases directly (see figure 1).

3.2 Semantic Query Language

To perform query at the Semantic Layer, we develop a Semantic Query Language, Query3 (Q3), which is designed specially for formulating query on databases and accurately captures the semantics of queries. Every Q3 query can be viewed as an OWL class definition; and query processing is reduced as computing instances satisfying the query concept definition. Typically, users use Semantic Browser to visually construct a Q3 query and then submit it to the Semantic Query Service of Dart-Grid for query processing. The set of statements in figure 2 is a query about the *name*, *usage*, *dosage* and *composition* of a *Chinese medical formula*, which can attend the disease of *influenza*.

```
q3:query
  q3:pattern [
    a tcm:Chinese_medical_formula;
    tcm:name [];
    tcm: usage_and_dosage [];
    tcm: composition [];
    tcm:attend [ a tcm: disease;
                  tcm:name "influenza";
                  tcm: pathogenesis [];
                  tcm: symptom_complex []
                ]
  ].
```

Fig. 2. An example of Q3 statements

The whole BNF syntax document about Q3 and more technique details can be referenced by visiting our website, <http://grid.zju.edu.cn/index.htm>.

3.3 Visual Semantic Mapping

The Q3 query statements above can be visually constructed based on semantic graphs and there is a direct semantic mapping from SGL to Q3 (see table 1):

Table 1. Mapping from Q3 to SGL and relevant operations

<i>SGL Element</i>	<i>Mapping Operation</i>	<i>Q3 BNF</i>	<i>Item Example</i>
		Operator	q3:query
sgl:graph	initialization	Pattern	q3:pattern
sgl:subgraph	select	blank_node	[...]
sgl:root	select and display	verb object	a tcm: Chinese_medical_formula
sgl:arc	select / select and display	Prop	tcm:name
sgl:node	select / select and display	Node	[]
sgl:node	input constraint	Literal	"influenza"

For expert users familiar with Q3, they can directly write down Q3 statements in the Dynamic Query Interface (DQI) of Semantic Browser, while for ordinary users Q3 statements can be constructed dynamically by visual semantic mapping during the process of Semantic Browse. The vectographic components of semantic graphs offer four mapping operations, “select”, “select and display”, “unselect” and “input constraint” (see figure 3). When end-users perform one of the operations at a semantic graph component, a corresponding Q3 item will be automatically produced or updated in the DQI. In this way, end-users can directly construct Semantic Query just by a group of sequential interactions with a visual semantic view, without knowing the structure or location of database resources.

3.4 Depth Control

The depth of each query is controllable and users can set the depth of semantic graphs by configuring parameters in Semantic Browser.

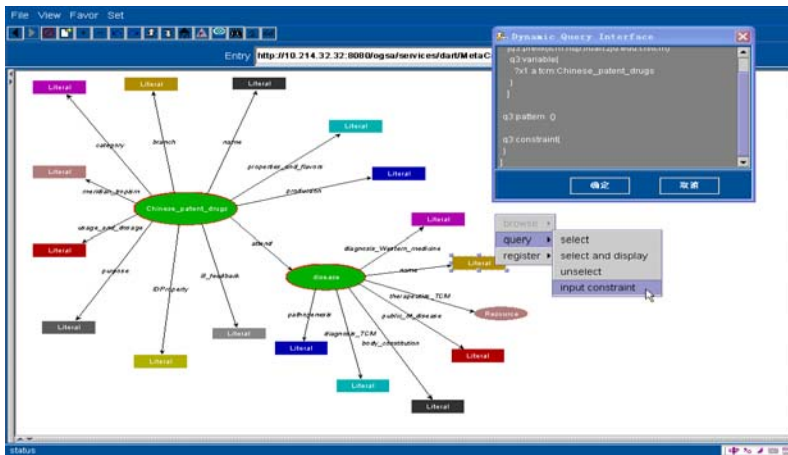


Fig. 3. Displaying the Semantic Query result with depth 2

Definition 4. *Display depth* is the depth in which semantic graphs is displayed. *Query depth* is the depth in which a semantic query is performed. *Slide Count* is the minimal times a user must take to browse the whole Semantic Query result. $Slide\ Count = Query\ depth / Display\ depth$.

4 Conclusion

In collaboration with the China Academy of Traditional Chinese Medicine, we have built a TCM information-sharing platform based upon Semantic Browser with Dart-Grid, which involves tens of large databases in many universities and research institutes. TCM researchers and doctors can gain valuable information by constructing Semantic Query visually in Semantic Browser, without caring about the locations and schemas of TCM database resources. The demo of our work can be found at <http://grid.zju.edu.cn/index.htm>. In this paper, we draw out a paradigm of visually constructing Semantic Query for large-scale database resources in Database Grid. Semantic Browser dynamically builds an visual semantic view based on Grid Services for end-users to perform high-level interaction in a dynamic and open environment. This work takes an important step in tackling the problem of sharing large-scale information resources under the Grid environment.

References

1. I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int'l J. High-Performance Computing Applications, 2001.
2. S. Tuecke, K. Czajkowski, I. Foster et al: Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum Draft Recommendation, 2003.

3. Wu Zhaohui, Chen Huajun, Huang Lican et al: Dart-InfoGrid: Towards an Information Grid Supporting Knowledge-based Information Sharing and Scalable Process Coordination. CNCC, 2003.
4. I. Foster et al: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. tech. report, Glous Project.
5. Yuxin Mao, Zhaohui Wu, Huajun Chen: Semantic Browser: an Intelligent Client for Dart-Grid. ICCS, 2004.
6. Lassila, O., Swick R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999.
7. Xuezhong Zhou, Zhaohui Wu, Aining Yin et al: Ontology Development for Unified Traditional Chinese Medical Language System. Special issue "AIM in China" of the International journal of Artificial Intelligence in Medicine, mid-2004 (in press).
8. Berners-Lee, T.: What the Semantic Web can represent. W3C, 1998.
9. Peter F. Patel-Schneider, Patrick Hayes, Ian Horrocks: OWL Web Ontology Language Overview. W3C Recommendation, 2004.
10. Peter Eklund, Nataliya Roberts, Steve Green: OntoRama.: Browsing RDF Ontologies Using a Hyperbolic-style Browser. CW2002, pp.405-411, Theory and Practices, IEEE press, 2002.
11. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 2001.
12. Mao Yuxin, Wu Zhaohui, Chen Huajun: SkyEyes: A Semantic Browser for the KB-Grid. GCC, 2003

A Distributed Data Server in Grid Environment*

Bin Chen, Nong Xiao, and Bo Liu

School of Computer, National University of Defense Technology
410073 Changsha, China
chenbin_hpyy@tom.com

Abstract. This paper introduces a high-performance distributed data server called DRB (Data Request Broker). DRB is the core of GridDaEn system which is a general Data Grid middleware that can provide uniform access and management of distributed and heterogeneous storage resources. DRB provides most of the core functions of GridDaEn including its uniform access to all kinds of geographically distributed, heterogeneous storage resources from a uniform and virtual view, and its supporting simple and complicated coordination across multiple administrative domains to form a federation, etc. DRB has the capability of providing high-performance and federated data service for data-intensive applications and researches over wide area networks.

1 Introduction

In recent years, many large-scale scientific researches and applications have the increasing needs of high-performance and large-capacity analysis and processing of mass datasets or storage resources which are geographically distributed and heterogeneous, such as global climate simulation, nuclear simulation, etc. Traditional data management infrastructure can't satisfy such needs. The emerging technology of Data Grid [1, 2] provides an effective solution to this problem. Data Grid builds an infrastructure and constitutes a uniform and virtual environment for uniform data access, management and processing by integrating all kinds of storage resources distributed over networks, and shields the distribution and heterogeneity of underlying storage resources for users.

GridDaEn (Grid Data Engine) system, a general Data Grid middleware designed and implemented by us, provides high-performance uniform data access, management and coordinated processing of distributed and heterogeneous storage resources over wide area networks by using technologies like distributed multi-domain federated servers, metadata catalog, etc. The main components of GridDaEn system include client tools, DRB (Data Request Broker) server and MDIS (Metadata Information Server). DRB is the core of GridDaEn system and is also a middleware between users and the resources that users request.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 presents the structure and features of DRB. Section 4 discusses the main

* This paper is supported by National Natural Science Foundation of China (60203016), National 863 High Technology Plan (2002AA131010) and 973-2003CB316900.

components of DRB. Section 5 gives some performance data of DRB. Finally, section 6 provides a summary and future work.

2 Related Work

Data Grid technology is developing rapidly in recent years. The Globus [7] system provides the capabilities of data access, movement and high-speed transfer by using GASS [3] and GridFTP [4]. It provides a better infrastructure for the development and application of Data Grid. The famous European Data Grid project [8] is based upon most of the basic services provided by Globus and aims to build the next generation computing infrastructure providing intensive computation and analysis of shared large-scale databases across widely distributed scientific communities. The SDSC Storage Resource Broker (SRB) [5] is a Data Grid middleware that supports uniform data access in distributed and heterogeneous storage environments. DRB is similar to the SRB server in SRB system. They all achieve uniform access to distributed, heterogeneous storage resources, which is based on data attributes and/or logical names rather than their names or physical locations. The latest version of SRB called ZoneSRB supports federation of multiple MCAT (Metadata Catalog) Zones. A zone in SRB system is controlled by a single MCAT and can include more than one SRB server. But each domain in GridDaEn system is controlled by a single DRB and can have one or no MDIS. DRB also has greater coordination capability to support federation of multiple domains and thus supports cross-domain and multi-domain federated data operations. DRB does not support container operation supported in SRB. The cost of maintaining container may be high and it's more suitable for system with tape such as HPSS. DRB uses data compression method to improve the efficiency of data transfer when large numbers of little-sized files are requested at a time.

3 DRB System Structure and Features

3.1 DRB System Structure

DRB is in the intermediate layer of GridDaEn system the architecture of which can be referred to [6], and achieves most of the core functions of GridDaEn system. The system structure of DRB is shown in Figure 1. The following gives a simple description.

- Security Service Layer is the entry to all kinds of data services DRB provides for users or other DRBs. Since the security issue is complex in grid environment, we'll not discuss it in detail in this paper.
- Data Service Layer includes some high-level services DRB provides for users or other DRBs such as file access services, cache management, replica management, etc, most of which are based upon undermentioned low-level services.
- Uniform Access Layer defines a set of high-level data access interfaces which are not relevant to specific underlying storage system.
- Resource Access Layer includes all kinds of underlying storage system access interfaces such as file system access interfaces, database access drivers, and so on.

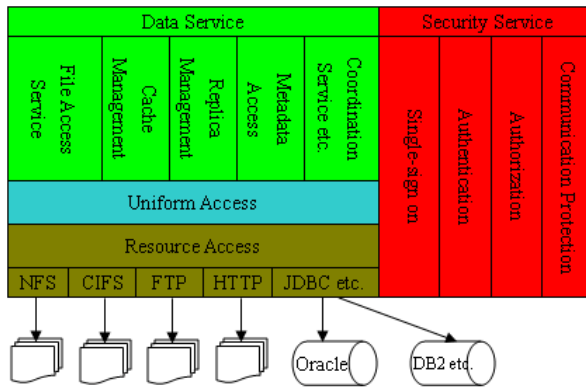


Fig. 1. The System Structure of DRB

3.2 DRB Features

1. **Uniform Access:** The client of DRB presents a uniform and virtual view to users from which DRB achieves the uniform access. DRB shields the underlying details of data resources such as physical location, physical file name, access protocol, etc.
2. **Multi-domain Federated Data Service:** The storage resources managed by Grid-DaEn system are organized into multiple domains each of which is controlled by a DRB. A DRB is an autonomous server. The coordination of multiple DRBs can achieve the federation of multiple domains to provide complicated data services.
3. **Multiple Access Modes:** Four types of access mode are supported in DRB to support access to storage resources that only have internal network addresses.
 - **CIALD:** User connects to DRB from intranet and accesses storage resources managed by the connected DRB, i.e. local domain.
 - **CIACD:** User connects to DRB from intranet and accesses storage resources not managed by the connected DRB.
 - **CEALD:** User connects to DRB from Internet or external networks and accesses storage resources managed by the connected DRB, i.e. local domain.
 - **CEACD:** User connects to DRB from Internet or external networks and accesses storage resources not managed by the connected DRB.
4. **Notification-Supported Data Service and Event-Driven Developing Model:** DRB provides rich command line tools, APIs and SDK for secondary development. The APIs include two types, namely synchronous APIs and asynchronous APIs, and support notification and event mechanism which are very useful and convenient for programmers to develop flexible, friendly and stronger data grid applications.

4 DRB Main Components

4.1 Uniform Access

DRB's capability of uniform access to distributed, heterogeneous storage resources is achieved by the uniform and virtual view which embodies the concept of "virtual

data”[9], and by defining uniform data access interfaces and encapsulating various underlying data access protocols such as NFS, CIFS, FTP, etc.

The uniform data access interfaces in DRB define a set of abstract high-level data access methods which are not relevant to underlying storage system. The interfaces separate DRB from implementation of specific storage system access interfaces and achieve plug and play (PnP) of heterogeneous storage systems. Supporting of access to other storage systems can be easily achieved by conforming to DRB’s uniform access interfaces. And high-level data services in DRB need no or little modification.

4.2 DRB Internal Structure

The main modules inside DRB include DRB Master, DRB Proxy, Global Scheduler, Cache and Replica Management, Data Transfer, etc, as illustrated in Figure 2.

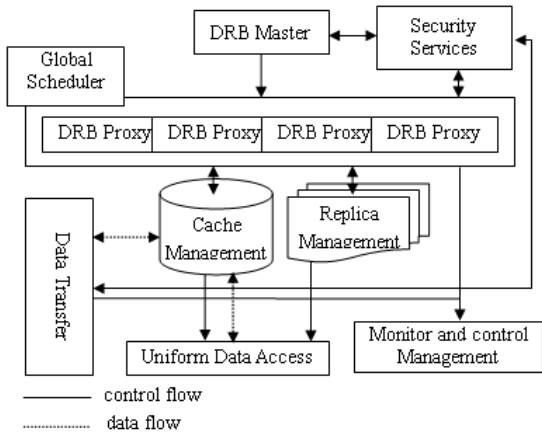


Fig. 2. The Sketch Map of DRB Internal Structure

DRB Master is a daemon thread, which monitors its well-known port for connections from users. Once a user connects, it authenticates the user. If the user passes the authentication, DRB Master generates a DRB Proxy thread to serve the user.

DRB Proxy is a thread that actually serves the user. If the resources are not within local domain, DRB Proxy will transmit user’s requests to remote DRB that manages the resources. The remote DRB will also generate a DRB Proxy to serve the user.

DRB supports concurrent accesses initiated by large numbers of users. The Global Scheduler module is in charge of scheduling users’ requests so as to improve the efficiency of data services. The default scheduling policy is FIFO. DRB administrator can also choose other policy such as Less Data with Higher Priority (LDHP), etc.

Other modules will not be discussed here in detail for the sake of brevity. The detailed information of these modules can be referred to [6].

4.3 Data Service in DRB

Users can access data resources managed by current connected DRB through the local-domain data service provided by DRB, and can also achieve data access across multiple domains through DRB’s multi-domain federated data service.

4.3.1 Local-Domain Data Service. DRB's local-domain data service provides a set of basic data access services such as file reading, writing, creation, deletion, etc. The detailed process of local-domain data service can be referred to the part "Data Access and Management" in [6].

4.3.2 Multi-domain Federated Data Service. The multi-domain federated data service is based upon the local-domain data service. If the data requested is not within local domain, DRB will access data across domains, which involves coordination of multiple DRBs. This coordination can be classified into two types, namely, simple coordination and complicated coordination.

- Simple Coordination

This type of coordination usually involves two DRBs. We assume user A connects to DRB A but requests data managed by DRB B. The process is illustrated in Figure 3.

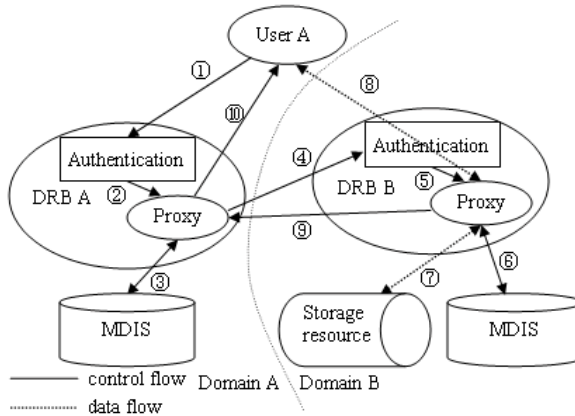


Fig. 3. Simple Coordination of DRBs

1. The process of connection establishment, authentication and generation of DRB Proxy is the same with the local-domain data service in section 4.3.1.
2. DRB A finds out the data requested is managed by DRB B and then it transmits the request of user A to DRB B.
3. DRB Proxy of DRB A is waiting for return message from DRB B.
4. DRB B receives request of user A from DRB A and executes the same procedure with that in the local-domain data service in section 4.3.1.
5. DRB A receives processing results from DRB B and also returns them to user A.

From the above process, we can see that simple coordination is essentially converting cross-domain access into local-domain access by transmitting user's request.

- Complicated Coordination

This type of coordination involves at least two DRBs. we assume that there are three DRBs called A, B, C and user A connected to DRB A wants to replicate a file named Example.txt from domain B controlled by DRB B to domain C controlled by DRB C. DRB uses an algorithm called Operation Decomposition and Time Division (ODTD)

to accomplish the complicated coordination. The ODTD algorithm decomposes a complicated operation into some basic operations that can be accomplished through the local-domain data services, and accomplishes them in different phases. The algorithm is described as follows and the process is illustrated in Figure 4:

1. Replication operation can be decomposed into three basic operations:
 - 1) Replication Metadata of Source File.
 - 2) Open File: DRB B reads file Example.txt into its server-side cache.
 - 3) Close File: DRB C writes back the “modified” file Example.txt from DRB B’s cache to destination physical storage resource managed by DRB C.
2. The whole replication process will be divided into three phases and a phase flag will be added to user’s request. Initially, the phase value equals 0. For convenience, we call the DRB which initiated the operation initial DRB, call the running DRB current DRB, call the DRB which manages the source file source DRB, and call the DRB which manages the destination file destination DRB:
 - 1) Phase 0: The phase value equals 0. The initial DRB replicates metadata of source file and sets phase flag to 1. The initial DRB is also the current DRB.
 - 2) Phase 1: The phase value equals 1. The current DRB finds out the source DRB. If current DRB is the source DRB, the Open File operation will be performed and phase flag will be set to 2. Otherwise, request will be transmitted to the source DRB and then the source DRB becomes the current DRB.
 - 3) Phase 2: The phase value equals 2. The current DRB finds out the destination DRB. If the current DRB is destination DRB, the Close File operation will be performed and phase flag will be set to 3. Otherwise, request will be transmitted to destination DRB and the destination DRB becomes the current DRB.
3. If phase value equals 3, the replication process is over and result is returned to user A. Each DRB executes the same procedure after receiving request from user A or from other DRBs, but takes different actions according to the phase value.

The red bold broken lines show the combination of the Open File operation and the Close File operation. For other special situations such as DRB A is source DRB or destination DRB, or is both source DRB and destination DRB, the processing is uniform and makes the most of the local-domain data services provided by DRB. For other more complicated operations such as third-party data movement, etc. DRBs can accomplish them well in a uniform way through the coordination of multiple DRBs if the operation decomposition and time division are appropriate.

5 Performance

Our experiment deployed two DRB servers in Institute of Computing Technology (ICT) of Chinese Academy of Science (CAS) and one DRB server in Tsinghua University. The client was deployed in National University of Defense Technology (NUDT). We mainly tested DRB’s local-domain data service and the multi-domain federated data service. The results of the experiment are illustrated in Figure 5.

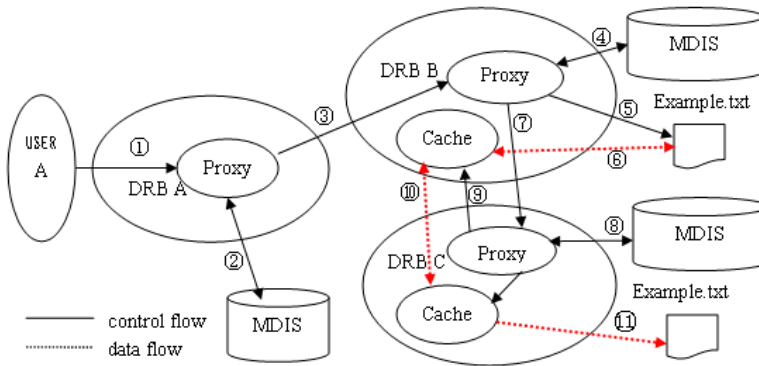


Fig. 4. Complicated Coordination of DRBs

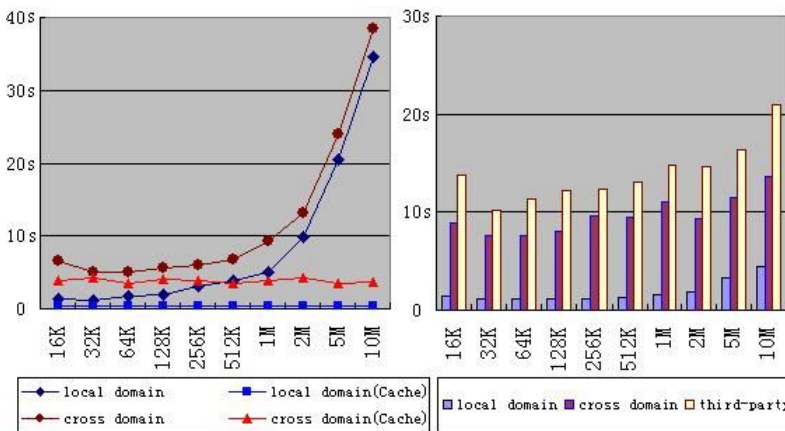


Fig. 5. DRB Reading Performance (left) and Replication Performance (right)

1. The cross-domain access always costs more time than local-domain access for extra cost of communication between DRBs.
2. The cache has significant influence on performance. When file was cached, the time spent on file reading was almost not relevant to file size.
3. The time spent on file access is usually increasing with the increasing of file size except the file sized 16K. The reason was that the first reading or replication operation needed some extra cost such as connection establishment, authentication.
4. The third-party replications of big files were faster than local-domain file reading, which were unexpected. The reason may be that the replications were performed between ICT and Tsinghua University where there was wider network bandwidth.
5. The access mode in our test was CEALD or CEACD which was not the most efficient. But DRB still presented better performance. Especially when wider network bandwidth was available, the federated data service presented higher performance.

6 Summary and Future Work

DRB is the core of GridDaEn system. Most of the data accesses are accomplished by DRB's local-domain data service or multi-domain federated data services. DRB takes many measures such as multi-level distributed cache management, metadata buffering, customized I/O policy, etc to improve DRB's performance, and provides well supporting of high performance and uniform access to distributed, heterogeneous mass storage resources. The performance enhancements of DRB will be our main research directions in the future, such as the application of P2P technology in data sharing across domains, high-speed data transfer protocols, etc.

References

1. I. Foster, C. Kesselman, S. Tuecke.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, (2001),15(3). 200-222
2. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke.: The Data Grid: Towards Architecture for the Distributed Management and Analysis of Large Scientific Data Sets. *Journal of Network and Computer Applications*, (2001), 23: 187-200
3. J. Bester, I. Foster, C. Kesselman, J. Tedesco, and S. Tuecke.: GASS: A data movement and access service for wide area computing systems. In *Proceedings of the Sixth Workshop on Input/Output in Parallel and Distributed Systems*, pages 78–88, Atlanta, GA, May (1999). ACM Press
4. L. Chervenak, I. Foster, C. Kesselman *et al.*: Data Management and Transfer in High Performance Computational Grid Environments. *Parallel Computing Journal*, (2002), 28 (5): 749-771
5. C Baru, R Moore, A Rajasekar, M Wan.: The SDSC Storage Resource Broker. In *Proc. Of CASCON'98 Conference*, Toronto, Canada, (1998)
6. XIAO Nong, LI Dongsheng, FU Wei, HUANG Bin, LU Xicheng.: GridDaen: A Data Grid Engine. *2nd International Conference on Grid and Cooperative Computing*, (2003)
7. <http://www.globus.org/>
8. Wolfgang Hoschek, Javier Jaen-Martinez, Asad Samar, Heinz Stockinger, and Kurt Stockinger.: Data Management in an International Data Grid Project. <http://www.eu-datagrid.org/>, (2000)
9. Jens-S. Vöckler, Mike Wilde, Ian Foster.: The GriPhyN Virtual Data System. Technical Report GriPhyN-(2002)-02

Author Index

- Bao, Hujun 728
Bao, Xiuguo 286
Byun, Sung-Wook 625
- Cai, Jia-Mei 115
Cai, Zhiping 52
Catuogno, Luigi 514
Chainbi, Walid 579
Chang, Guiran 319
Chang, Zhi-ming 689
Chen, Bin 775
Chen, Deren 728
Chen, Fujie 753
Chen, Haitao 450
Chen, Huajun 148, 768
Chen, Jiu Jun 59
Chen, Shudong 156
Chen, Wei 704, 712
Chen, Xueyao 319
Choi, Eunmi 19
Cremer, James 720
- Dai, Yafei 213
Deng, Qian-ni 673
Deng, Shuiguang 27, 99
Dong, Xiaoshe 181
Dong, Zhao 712
Dou, Qiang 697
Dou, Wenhua 587
Du, Dehui 75
Duan, Hai-Xin 374
Duce, David 633
- Fan, Jing 115
Fang, Binxing 270, 286, 466, 530
Fang, Meiyu 148
Faruolo, Pompeo 514
Feng, Xu 430
- Gao, Chongnan 187
Gao, Ji 59
Gao, Yuan 571
Gao, Zhiqiang 83
Gong, Zhenghu 398, 450
Gu, Ning 140
Guangwen, Yang 36
- Gui, Xiaolin 482
Guo, Xiaosheng 164
- Han, Dezhi 195
Han, Jun 366
Han, Xiaoxu 720
Han, Yanbo 90, 107
Hao, Huang 430
He, Dake 329
He, Erhua 697
He, Keqing 67, 75
He, Yangfan 75
He, Yanxiang 609
Hu, Guangming 398
Hu, Haitao 90
Hu, Hanping 506
Hu, Hua-Ping 382, 398, 450, 458, 546
Hu, Jun 59
Hu, Mingzeng 270, 286, 530
Huang, Chang 11
Huang, Changqin 744
Huang, Chenlin 382
Huang, Chuanhe 609
Huang, Dazheng 36
Huang, Kui 90, 107
Huang, Xin 681
Huang, Zhengge 681
Huang, Zunguo 398, 414, 450, 458
Huebscher, Markus C. 597
Huo, Haitao 538
- Jang, Jongsu 406
Jayasena, Sajindra 303
Jeong, Chang-Sung 625, 736
Jia, Xiaohua 609
Jiang, Jie 587
Jiang, Sheng-Yi 422, 522
Jiang, Yichuan 335
Jiang, Zhongding 728
Jin, Shi-Yao 458
Jing, Ning 123
Jun, Tang 617
Jun, Xie 430
- Kim, Hyo 311
Kong, Tao 506

- Kuang, Li 99
 Kwon, Yong-Won 625

 Lee, Cheolho 390
 Lee, Jang-sun 311
 Lee, Tae-Dong 736
 Lee, Yun-Kyoung 625
 Lei, Yu 728
 Li, Baiyan 498
 Li, Bigang 262, 278
 Li, Bing 67
 Li, Chun-Jiang 689
 Li, Dongsheng 221
 Li, Gang 90, 107
 Li, Minglu 156, 358, 498, 650
 Li, Qing-Hua 422, 522
 Li, Sikun 753, 761
 Li, Xiaojian 538
 Li, Xie 430
 Li, Xing 374
 Li, Yinan 482
 Li, Ying 27
 Li, Yunchun 181
 Liang, HaiYing 571
 Liang, Peng 67
 Liao, Bei Shui 59
 Liao, Xiangke 343
 Lin, Chuan 99
 Lin, Xin-hua 673
 Liu, Bo 351, 775
 Liu, Da-Xin 3
 Liu, Gang 270
 Liu, Guangming 697
 Liu, Hong 673
 Liu, Hui 351, 358
 Liu, Jin 67
 Liu, Wu 374
 Liu, Xianghui 52
 Lu, Xicheng 52, 221, 546
 Lu, Xin-da 673
 Luan, Zhongzhi 181
 Luo, Yihui 238

 Ma, Chunbo 329
 Ma, Fanyuan 156
 Ma, Huadong 173
 Mao, Xinjun 633
 Mao, Yuxin 148, 768
 McCann, Julie A. 597
 Min, Dugki 19

 Moon, Jongsub 390
 Moon, Kiyong 406
 Mu, Fei 278

 Nagaraj, N.S. 642
 Natarajan, Swaminathan 642
 Ni, Jun 720
 No, Jaechun 311

 O'Hare, Gregory M.P. 658

 Pan, Guoteng 697
 Park, Namje 406
 Peng, Qunsheng 712
 Peng, Rong 67
 Peng, Yuxing 753, 761
 Petrillo, Umberto Ferraro 514
 Pinto, Colin 642
 Pooloth, Krishnakumar 642

 Qi, Xiao-Feng 673
 Qian, Depei 181, 482
 Qian, Qi 650
 Qin, Zheng 246
 Qin, Zhiguang 437
 Qiu, Yingjun 704
 Qu, Min 213
 Qu, Xiangli 44, 554, 689
 Qu, Yuzhong 83

 Rao, Ruonan 498
 Ren, Bo 115
 Ren, Ping 374
 Ryu, So-Hyun 625

 Schikuta, Erich 132, 563
 See, Chong Wee 303
 Seo, Jungtaek 390
 Shan, Lijun 633
 Shen, Jianhua 156
 Shen, Ling 538
 Shen, Meiming 294
 Shen, Song 658
 Shi, Jiaoying 704
 Shu, Ji-wu 203, 229, 254, 262, 278, 294
 Sindhgatta, Renuka 642
 Sohn, Sungwon 406
 Song, Guanghua 744
 Song, Jie 303
 Song, Shi-Jie 458
 Stoelwinder, Abele 303
 Sun, Weitao 254

- Sun, Xi-li 673
 Tan, Yubo 753
 Tang, Siliang 164
 Tang, Yong 546
 Tang, Yu 123
 Visconti, Ivan 514
 Wang, DaDong 571
 Wang, Di 294
 Wang, Dongsheng 187
 Wang, HongJun 571
 Wang, Huaimin 343
 Wang, Hui 422, 522
 Wang, Jing 107
 Wang, Lunwei 343
 Wang, Qiang 67
 Wang, RuiJun 571
 Wang, Wei 466
 Wang, Zhihong 720
 Wang, Zhiying 382
 Wei, Lifeng 343
 Weiqin, Tong 617
 Weishäupl, Thomas 132, 563
 Wen, Dong-chan 229
 Wong, Wai-Hong 303
 Wu, Huayi 609
 Wu, Jian-Ping 374
 Wu, Wei 538
 Wu, Xiaogang 506
 Wu, Xiaojun 148
 Wu, Yongwei 36
 Wu, Yujin 140
 Wu, Zhaohui 11, 27, 99, 148, 164, 768
 Xiang, Guang 44, 554
 Xiao, Nong 44, 221, 775
 Xiaoli, Zhi 617
 Xie, Bing 482
 Xie, Changsheng 195, 238
 Xie, Lijun 681
 Xie, Luoguo 697
 Xiong, Huanyu 422
 Xu, Chuanfu 450
 Xu, Jiefeng 246
 Xu, Xin 414, 490
 Xu, Zhao 148
 Xuan, Lei 414, 490
 Yan, Lu 474
 Yan, Rui 203, 229
 Yang, Jianhua 164
 Yang, Wei 319
 Yang, Xue-Jun 44, 554, 689
 Yao, Yuanzhe 466
 Yee, Chin-Peng 303
 Yi, Faling 195
 Yin, Jianping 52
 Ying, Shi 75
 Yoo, Seung-Hun 736
 You, Jinyuan 498
 You, Xindong 319
 Yu, Bing 203
 Zeng, Wandan 319
 Zhai, Yuqing 83
 Zhang, Baoliang 506
 Zhang, Chengfeng 238
 Zhang, Feng 437
 Zhang, Haiyang 173
 Zhang, Hongli 270, 466, 530
 Zhang, Jifa 681
 Zhang, Long 712
 Zhang, Shaohua 140
 Zhang, Shiyong 335
 Zhang, Wan-Song 3
 Zhang, Wenju 156
 Zhang, Xingjun 181
 Zhang, Yongjun 761
 Zhang, Youhui 187
 Zhao, Jianmin 52
 Zhao, Yang 213
 Zhao, Youbing 704
 Zhao, Zhuofeng 90
 Zheng, Guozhou 11, 148
 Zheng, Jun 530
 Zheng, Weimin 36, 187, 254, 262, 278
 Zheng, Yao 681, 744
 Zhijie, Yin 617
 Zhong, Yiping 335
 Zhou, Bin 633
 Zhou, Haifang 554, 689, 697
 Zhou, Shijie 437
 Zhou, Zhaotao 107
 Zhu, Hong 633
 Zhu, Xiaoqian 697