Resource Management and Organization in CROWN Grid

Jinpeng Huai[†] , Tianyu Wo[†], Yunhao Liu[‡]

[†]*Dept. of Computer Science and Technology, Beihang University*
[‡]*Dept. of Computer Science, Hong Kong University of Science & Technology*

Abstract— The main goal of the key project, CROWN Grid, is to empower in-depth integration of resources and cooperation of researchers nationwide and worldwide. CROWN project was started late 2003 and we have successfully released CROWN 2.0 in November of 2005. In this paper, we introduce the resource management and organization in the CROWN grid.

Keyword: **GROWN, grid computing, service club, trust, incentive**

1. Introduction

Grid computing enables users to share resources, including CPU, storages, memory, bandwidth, etc, across multiple domains [7, 3, 8, 12, 11] over wide-area network. The main goal of our key project, CROWN (**C**hina **R**&D **E**nvironment **O**ver **W**ide-area **N**etwork), is to empower in-depth integration of resources and cooperation of researchers nationwide and worldwide. CROWN was started in late 2003. Tens of universities and institutes, such as Tsinghua University, Peking University, China Academy of Sciences and Hong Kong University of Science and Technology, Beihang University from cities have joined CROWN.

Through the Computer Network Information Centre of CAS, the CROWN is connected to some famous international grid testing bed such as GLORIAD and PRAGMA. Lots of applications in different domains have been deployed into CROWN grid, such as gene comparison in bioinformatics, climates pattern prediction in environment monitoring, etc.

Along with the growth the size of a grid system, resource management and organization become a challenge issue in grid systems. Obviously, a centralized mechanism greatly suffers single point of failure. Hence, an efficient resource management mechanism for a highly heterogeneous, dynamic distributed environment is of great importance, and there indeed have been many researchers worked in resource management approaches such as Buyya et al. [17, 19, 2].

In this paper, we focus on the issue of maximizing resource utilization and providing dynamic sharing and collaboration mechanisms in GROWN grid. The rest of this paper is organized as follows. We introduce the architecture of CROWN system in Section 2. Sections 3 and 4 discuss the resource organization and management mechanism in static and dynamic grid environments, respectively. Section 5 concludes the paper.

2. Architecture of CROWN

2.1 Overview

CROWN employs a multi-layered structure of resource organization and management, as illustrated in Fig. 1, based on the characteristic of e-Science applications and the resource subordination relationship. It includes four major entities:

- **Node:** All kinds of heterogeneous resources are encapsulated into CROWN nodes, and services are deployed on these nodes to provide a homogeneous view for upper middleware to access the resources.

- **Domain:** Nodes are organized into different domains according to their subordination or resource catalogs. A Resource Locating and Description Service (*RLDS*) is deployed in each domain as the center of grid information management. Domains may contain several sub-domains such that all RLDS come up with a tree like topology.
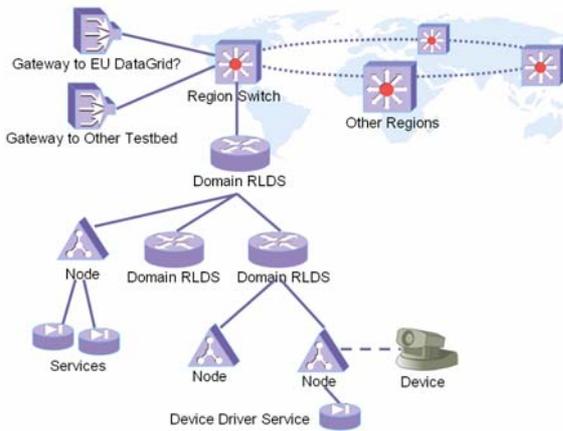
Fig 1: CROWN resource organization and management

- **Region:** For flexibility, the concept *region* is introduced to coordinate different domain trees. A region switch is deployed in each region such that flexible topologies and information sharing mechanisms can be applied among regions.
- **Gateway:** To connect with other grid systems, we deploy several gateways on the edge of the CROWN system.

2.2 CROWN Middleware

CROWN middleware is built on top of physical resource layer which is consisting of existing heterogeneous resources. A rich set of software components and tools are provided in the middleware layer to support the development and running of grid services in the environment. An application layer is built based on the middleware layer of CROWN providing grid applications for multi-discipline e-Science research, as shown in Fig. 2.

CROWN is a service oriented grid application supporting environment. Resource sharing and collaboration are achieved through grid services. A grid service container, called NodeServer, is deployed on each of the grid hosts. All the services are encapsulated as certain types of resources, with some of them providing application specific functions and others providing general services, such as grid information services (GIS).

In CROWN, before a computer becomes a Node Server (NS), it must be installed with CROWN middleware. The service container is the core component in CROWN middleware, which provides a runtime environment for various services. Each NS usually belongs to a security domain. Every domain has at least one *RLDS* to provide information services, and *RLDS* maintains the dynamic information of available services.
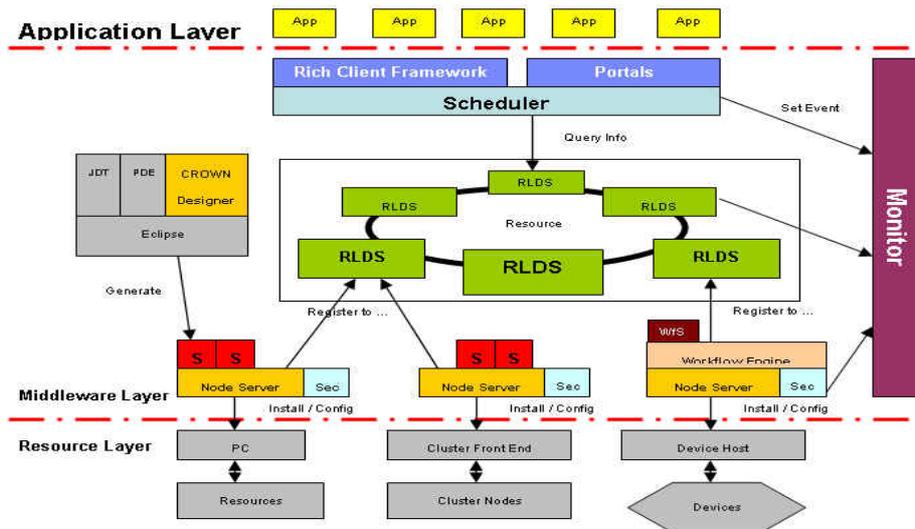

Fig 2: Architecture of CROWN middleware

By querying RLDS for information of available grid services, the *Scheduler* can select proper resource to execute the job and solve the problem on behalf of the grid user. CROWN Designer is an IDE for service developer. The CROWN Monitor is to trace the system status in a global view and help to analyze the system running behaviors. A full-fledged grid security solution is also provided in CROWN which contains message level security, authentication and authorization, credential management, and identity mapping for heterogeneous security infrastructure and trust negotiation.

2.3 Testbed and Applications

A CROWN testbed is established based on existing network infrastructure (CERNET and CSTNET). Several pilot applications are running in the CROWN environment.

- **AREM:** The **A**dvanced **R**egional **E**ta-coordinate numerical prediction **M**odel is to meet the requirement of meteorology and atmospheric physics researchers to analyze weather and visualize the result and refine prediction models.

- **MDP:** Combining speech recognition engine with grid technology, a massive **M**ultimedia **D**ata **P**rocessing platform is provided to support parallel multi-way speech recognition.

- **gViz:** A visualization tool for simulation data developed by University of Leeds. After encapsulating it into grid service and deploying it in both CROWN and White Rose Grid in UK, a large number of users can make use of this function simultaneously.

- **DSSR:** **D**igital **S**ky **S**urvey **R**etrieval application is a tool for searching and retrieving a whole star graph according to certain regional parameters. It helps astronomical researchers to do further observation, analysis, and studies in an efficient way.

3. S-Club: Resource Organization in Static Environments

Early computational grids employed the layered architecture with an "hourglass model" [8]. Recently, with the evolution of Web services, the service-oriented architecture has become a significant trend for grid computing, with OGSA/ WSRF as the de facto standards [9, 10]. CROWN has adopted the service-oriented architecture, connecting large amount of services deployed in universities and institutes.

Many efforts have been made to improve information service for grid computing. Some centralized approaches, such as MDS-1 [7], Hawkeye [1] and R-GMA [2], have been designed for small-scale grid systems. In these approaches, only one centralized server is deployed to provide information service.

To provide information service for large scale grid systems, several solutions, such as MDS-2 [6] and GAIS [15], have adopted a hierarchical architecture to organize GISs. To search services over the GISs, however, we have to traverse a partial of the hierarchical architecture, which could cause long latency as well as huge search traffic.

CROWN employs a service club mechanism, called S-Club, for efficient resource organization and service discovery.

3.1 Design Rational

The basic idea of S-Club is to build an overlay over the existing GIS mesh network. In such an overlay, GISs providing the same type of services organized into a service club. An example of such a club overlay is shown in Fig. 3, where nodes C, D, E, G form a club. A search request could be forwarded to the corresponding club first such that search response time and overhead can also be reduced if the desired result is available in the club.
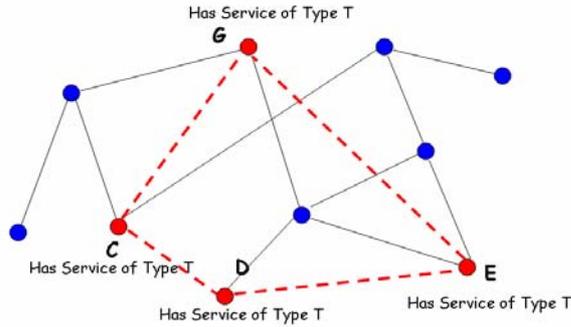
Fig 3: An example of service club

Fig 4: Average response time

Intuitively, to set up a club requires information exchange, and clubs need to be maintained dynamically because new GISs may join and some existing GISs may leave. Also, it is possible that some types of services become less popular after the club is built. Therefore, we have to be careful on the trade-off between the potential benefit and the cost incurred. In general, the usage of services is not uniformly distributed. Some types of services can be very popular and others may not. When/how clubs are constructed/destroyed will be key issues in S-club scheme.

We assume that any search request is firstly sent to a GIS close to the user. On receiving a search request for a specific service type, the GIS checks locally whether there has been a club for this type. If yes, the GIS forwards the request to the club, which will be flooded within the club only. If there is no club for this type, however, the GIS floods the request throughout the mesh network.

When a new GIS joins the GIS network, it has no idea what clubs are there. But since it has at least one neighbour in the underlying mesh network, it can ask one of its neighbours for the information of existing clubs. Namely, it simply copies the information of clubs from its neighbour. For more detailed discussion, please refer to [13].
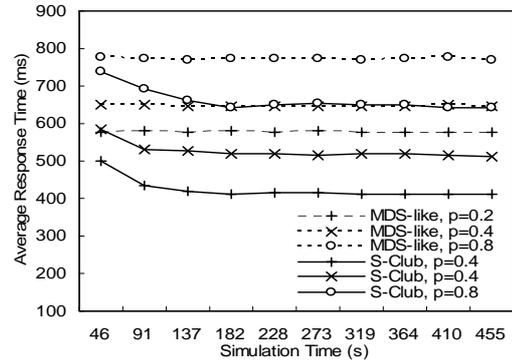
3.2 Performance

In order to evaluate S-Club, we use following three performance metrics: average response time, total traffic overhead, and optimization ratio.

Average Response Time of a query is one of the parameters concerned by end users. We define response time of a query as the time period from when the query from end user is received by the first GIS node, and until when this node collects all the response results from other GISs and meets the end user's demands.

Total Traffic Overhead of a query is defined as the length of message sent among GISs to answer the query. If S-Club is adopted, the traffic overhead of creating, maintaining and deleting is also included. We have traced the real message in CROWN Grid, in which GIS is wrapped as an OGSA Grid Service using SOAP messages. Including the head of SOAP message, the size of all related messages is ranging from 1500 bytes to 3000 bytes.

Optimization Ratio shows the benefit of S-Club mechanism with comparison to MDS-like system. The optimization ratio of average response time is defined as

$$OptimizationRatio = \frac{ART_M - ART_S}{ART_M}$$

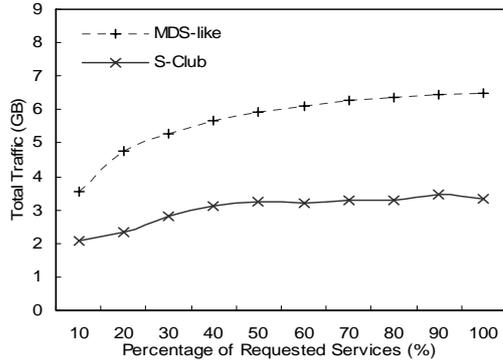Where ART_M is the average response time of MDS-like system and ART_S is the average response time of S-Club system.
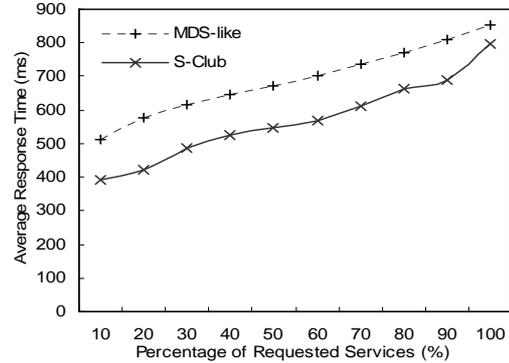
Fig 5: Total traffic v.s. % of requested services

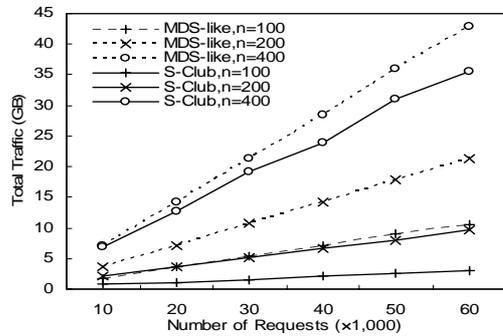

Fig 6: Average response time v.s. % of requested services

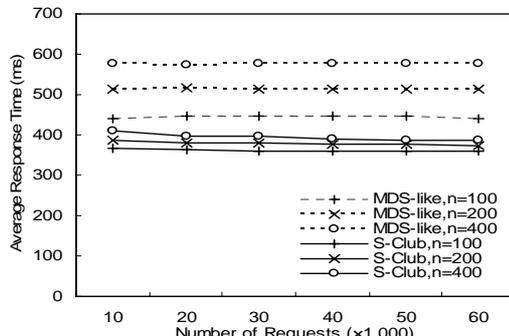**Fig 7: Total traffic v.s. number of requests**

Fig 8: Response time v.s. number of requests

We compare the S-Club with MDS-like [6] systems. In Fig. 4, the parameter p stands for the percentage of requested services. We see that with the time elapses, average response time of all MDS-like systems is almost a constant while the average response time of S-Club systems is decreasing gradually and finally comes to a stable state.

Figures 5 and 6 plot the benefit S-Club mechanism brings with the change of parameter p with a GIS network of 400 nodes. With the increment of p, total traffic and average response time increase together. We find that S-Club reduces the total traffic by 46% and average response time by around 7-26%.

The experimental results in Fig. 7 and Fig. 8 show that S-Club reduces total traffic and average response time significantly. For example, when n=200 and total number of requests is 60000, S-Club reduces traffic by 55% and average response time by 27%.

Compared with the previous approaches, S-Club adopts a fully decentralized architecture with an unstructured topology. Each GIS keeps the information of services registered to it. To improve the performance, overlay is constructed dynamically and may be changed constantly with the self-organizations of service clubs.

4. Resource Organization in Dynamic Environments

In S-Club, we assume the grid nodes are not selfish and would like to share their resources with others. In order to tackle the free riding problem, CROWN combines the soft incentive [16, 5, 15] and hard incentive schemes [1, 4, 19] and propose a trust-incentive framework.
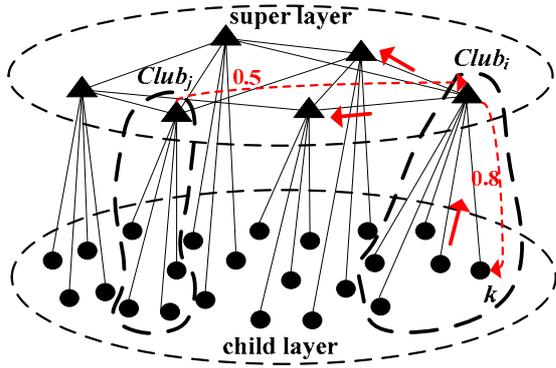
Fig 9: S-Club based two tier organizations

4.1 Basic idea

We borrow the principles of market pricing when constructing the trust and incentive model. The primary goals are securing shared resources, promoting users to share valuable resources, maintaining the balance of supply and demand in competitive grid resource market, and finally maximizing aggregate resource utilization.

As illustrated in Fig. 9, the distributed resource management in CROWN grid adopts a two-tier peer-to-peer architecture [18]. The super layer is the backbone consisting of S-Clubs; the child layer includes all clients and resource providers.

Nodes may join and leave the collaboration dynamically, or transfer from one club to another club. Thus, the super layer nodes may have the trust records for child nodes. In this approach, the more resources a node provides, the higher degree of trust a node has. Obviously, every node is willing to cooperate with a node with higher trust value.

The allocation mechanism is as follows.

- In each period of bidding, we calculate the per unit valuation $bid_j^i = \dfrac{V_j^i}{Q_j^i \times et_j^i}$ for each bid in Clubj, where V_j^i, Q_j^i, et_j^i denote bidding price, node number, and estimated execution time, respectively. Then we use maxbid to denote $max\left[bid_j^1, bid_j^i, \cdots, bid_j^k\right]$.

- We calculate the evaluation value of each bid by the formula:

$$evlbid_j^i = \alpha \times \frac{bid_j^i}{maxbid} + (1-\alpha) \times trust_{cid}^{pid},$$

where $trust_{cid}^{pid}$ is the trust value from Clubj to $Child_{cid}^{pid}$, and $\alpha \in [0,1]$ is risk degree of Clubj according to the benefit and secure factors. If Clubj focuses more on benefit than security, set α bigger; otherwise, set α smaller.

- We sort all bids in descending order according to $evlbid_j^i$.

- We go through the sorted bid list and evaluate each single bid. If the resource request for a bid can be fulfilled with the remaining node resources, we will allocate the resources to the bid.

4.1 Performance

We conduct an experiment with a set of five peers with varying currency and trust value, bidding for some portion of 200 unit resources. In the experiment, the amount of resource that each bid i can obtain is determined by $\dfrac{TotalResources \times evlbid^i}{\sum_i evlbid^i}$, where $evlbid$ is the evaluated value of a bid. We adopt this distribution policy to protect light users against starvation from heavy users when the demand is over the supply.

We define a metric named *allocation_ratio* for each bid as follows.

$$allocation\_ratio = \frac{allocation\_quantity}{total\_request\_quantity}$$

In this experiment, we assume that the total request quantity of each node is 100 units. There are five curves in Figures 10 and 11, where x-axis represents the bids times, and y-axis represents the allocation ratio. We first set the risk degree $\alpha = 1$ in Fig. 10 and consider the security factor and set the risk degree $\alpha = 0.2$ in Fig. 11.
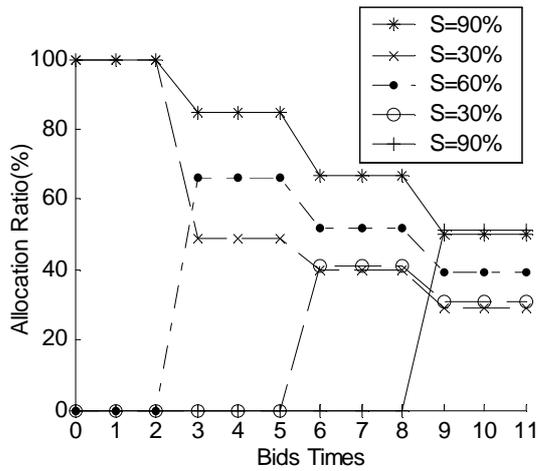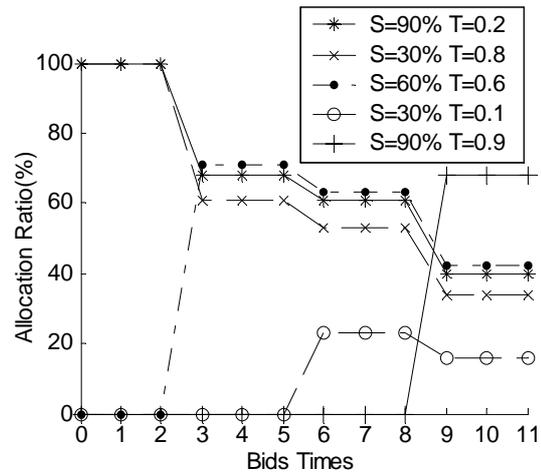
Fig 10. Incentive compatible allocation

Fig 11.Trust incentive allocation

The symbol *S* in both figures denotes the ratio of providing resources, for example, the first node provides 90% of local resources to other nodes. The symbol *T* in Fig. 11 denotes the trust value of bidding nodes. In the first three periods, only two nodes request for 100 units resources and the supply meets the demand. Thus the allocation ratios for the two nodes are all 100%. After the third period, the increasing bids outnumbered the supply.

As shown in Fig. 10, without the security consideration, the nodes providing the same resources nearly obtain the same allocation ratio, and the more resources a node contributes, the higher allocation ratio a node obtains. However, if considering the trust value in Fig. 11, we can see that the provision ratio of node 3 (60%) is less than that of node 1 (90%), but node 3 has a higher trust value (*T*=0.6) and thus obtains more resources than node 1. Similarly, node 2 and node 4 have the same provision ratio: 30%. Node 2 obtains more resources than node 4 at the sixth bid in Fig. 6, because the trust value of node 2 (*T*=0.8) is far greater than that of node 4 (*T*=0.1).

Thus, by evaluating the allocation scheme, we see that peers can obtain more resources only having shared more resources and accumulated higher degree of trust.

5. Conclusion

In order to maximize resource utilization as well as providing trust management in grid computing environments, we discuss two resource organization and management mechinism adpoted in CROWN grid. By building a service club based overlay over the existing mesh network of GISs, we improve the resource locating efficiency. By introducing the price strategy, we encourage nodes to share more resources, and ensure the balance of supply and demand, as well as enhance the resource utilization of grid environment. We show the effectiveness of our approach through experiments in CROWN [14] system in our lab.

6. Acknowledgements

7. Reference

[1]A. AuYoung, B. N. Chun, A. C. Snoeren, and A. Vahdat, "Resource Allocation in Federated Distributed Computing Infrastructures," in Proceedings of the 1st Workshop on Operating System and Architectural Support for the on demand IT InfraStructure, 2004.

[2]R. Buyya, D. Abramson, and S. Venugopal, "The Grid Economy," *Special Issue on Grid Computing,Proceedings of the IEEE, Manish Parashar and Craig Lee (editors)*, vol. 93, pp. 698-714, 2005.

[3]S. J. Chapin, D. Katramatos, J. Karpovich, and A. Grimshaw, "Resource management in Legion," in Proceedings of Workshop on Job Scheduling Strategies for Parallel Processing, in conjunction with the International Parallel and Distributed Processing Symposium, 1999.

[4]B. N. Chun, J. A. C. Ng, D. C. Parkes, and A. Vahdat, "Computational Resource Exchanges for Distributed Resource Allocation," 2004.

[5]M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks," in Proceedings of ACM E-Commerce Conference (EC'04), 2004.

[6]S. Fitzgerald, I. Foster, C. Kesselman, G. v. Laszewski, W. Smith, and S. Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations," in *the 6th IEEE symposium on High-Performance Distributed Computing*, 1997, pp. 365-375.

[7]I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," in *International Journal of Super-computer Applications*, vol. 11, 1997, pp. 115-129.

[8]I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid," *International Journal of Super-computer Applications*, 2001.

[9]I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The physiology of the Grid: An Open Grid Services Architecture for distributed systems integration, http://www.globus.org/research/papers/ogsa.pdf

[10]I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "Grid Services for distributed system integration," in *IEEE Computer Magazine*, vol. 35, 2002, pp. 37-46.

[11]I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*: Morgan Kaufmann Publishers, 2003.

[12]J. Frey and T. Tannenbaum, "Condor-G: A computation Management Agent for multi-Institutional Grids," *Journal of Cluster Computing*, vol. 5, pp. 237, 2002.

[13]C. Hu, Y. Zhu, J. Huai, Y. Liu, and L. M. Ni, "Efficient Information Service Management Using Service Club in CROWN Grid," in Proceedings of IEEE International Conference on Services Computing, 2005.

[14]J. Huai, Y. Zhang, X. Li, and Y. Liu, "Distributed Access Control in CROWN Groups," in Proceedings of International Conference on Parallel Processing (ICPP), 2005.

[15]T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "A Game Theoretic Approach to Provide Incentive and Service Differentiation in P2P Networks," in Proceedings of ACM SIGMETRICS/PERFORMANCE, 2004.

[16]K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster, "To Share or not to Share: an Analysis of Incentives to Contribute in Collaborative File Sharing Environments," in Proceedings of Workshop on Economics of Peer to Peer Systems, 2003.

[17]R. Ranjan, A. Harwood, and R. Buyya, "Grid Federation: An Economy Based, Scalable Distributed Resource Management System for Large-Scale Resource Coupling," Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, 2004.

[18]L. Xiao, Z. Zhuang, and Y. Liu, "Dynamic Layer Management in Super-peer Architectures," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2005.

[19]C. S. Yeo and R. Buyya, "Pricing for Utility-driven Resource Management and Allocation in Clusters," in Proceedings of the 12th International Conference on Advanced Computing and Communication, 2004.