Performance Enhancement of Scheduling Algorithms in Clusters and Grids using Improved Dynamic Load Balancing Techniques

Hemant Kumar Mehta
School of Computer Science, Devi Ahilya University, Indore, India
+919425077901
mehtahk@yahoo.com

Priyesh Kanungo
Patel Group of Institutions, Ralamandal, Indore, India
+919406623744
priyeshkanungo@hotmail.com

Manohar Chandwani
Institute of Engineering and Technology
Devi Ahilya University, Indore, India
+919303227853
chandwanim1@rediffmail.com

ABSTRACT
This paper describes the research work done for during PhD study. Cluster computing, grid computing and cloud computing are distributed computing environments (DCEs) widely accepted for the next generation Web based commercial and scientific applications. These applications work around the globally distributed data of petabyte scale that can only be processed by the aggregating the capability of globally distributed resources. The resource management and process scheduling in large scale distributed computing environment are a challenging task. In this research work we have devised new scheduling algorithms and resource management strategies specially designed for the cluster and grid cloud and peer-to-peer computing. The research work finally presented the distributed computing solutions to one scientific and one commercial application viz. e-Learning and data mining.

Categories and Subject Descriptors
C.2.4 [**Computer-Communication Networks**]: Distributed Systems –*Distributed applications.*

General Terms
Algorithms, Design, Economics, Experimentation, Management, Performance

Keywords
Cluster, cloud, grid, resource management, grid service, trust management

1. INTRODUCTION
The popular web applications have observed explosive growth in terms of number of requests from different users. Now a days, various kinds of commercial and scientific applications are implemented as a web-based application or a web service application. Web services develop to be accessed from the various web applications etc. These different levels generate load on the websites at different levels from lower to higher respectively. Web services are web applications that use the hyper text transfer protocol (HTTP) for communication, similar to the web sites. These applications belong to the area of particle physics, life science, geo science, telecommunications, chemical & material science, financial applications, manufacturing, entertainment, media and gaming. Such applications either generate large amounts of data that entail extensive computations like data mining applications or they are computation extensive problems like life science applications. To meet the processing needs of such applications, various types of distributed systems have been evolved over a period of time. In the beginning, super computers were used by such applications. The set up and maintenance of the super computers are very costly. Cluster computing (*Cluster*) is a cost effective and fail-safe alternative to the super computers. *Cluster* is used if the processing has to be done within the same organization itself. Grid computing (*Grid*) shares the capability of resources in geographically distributed cluster of various organizations to form a single logical virtual organization (VO). The participation of organization in *Grid* is based on mutual collaboration, revenue or donation basis. *Grid* and *Clusters* can be used for market oriented computing as well as service oriented computing. In market oriented computing, consumers have to pay to the resource providers for the resources and services being used. Service oriented computing focuses on providing business processes in the form of Web services. The concepts of *Grid*, market oriented computing and service oriented computing evolved further and they are combined together to develop Cloud computing. Cloud computing (*Cloud*) is a collection of dynamically scalable and virtualized resources, which are provided as a service over the Internet. These services can be Software as a Service (*SaaS*), Platform as a Service (*PaaS*) and Infrastructure as a Service (*IaaS*).

WWW 2011, March 28–April 1, 2011, Hyderabad, India.

ACM 978-1-4503-0637-9/11/03.

2. PROBLEM ADDRESSED
This PhD research work has been done with the following specific objectives:

1. To implement the load balancing policies being used in *Cluster* to *Grid*. These policies have been initially developed for *Cluster* and needs some modifications so that they can be effectively used in *Grid*.

2. To develop a distributed version of the delay strategy proposed by Hui and Chanson that can work in multi domain environments including Cluster and Grid. Hui and Chanson have proposed dynamic load balancing strategy that delays the execution of newly arrived jobs when the system is under full utilization. This will improve the response time perceived by both the newly arrived and currently running processes [9].

3. To modify existing load balancing policies used in cluster environment to incorporate content awareness. It is also planned to implement content aware load balancing policies in a Grid environment.
4. Analysis of the existing dynamic content caching schemes to find their limits. This research also proposed to use dynamic content handling techniques for performance enhancement in Cluster and Grid.
5. To develop the data access and integration services that utilizes the grid infrastructure for database management applications. These services may be integrated with the existing grid environments.
6. To resolve the issues in *Grid* such as scalability, connectivity and resource discovery it is planned to utilize the similarities between P2P and grid so that P2P protocols and models can be exploited.

The first three objectives are based on the policy making for distributed scheduling, the fourth objective takes care of dynamic data caching and the fifth & sixth objectives are meant for developing a web service/ grid service based policy for data integration and resource registration.

3. STATE OF THE ART

Extensive literature review has been carried out in the area of scheduling algorithms and resource management in *Cluster* and *Grid*. The literature contains of large body of research work however, the discussion is limited to the selected topics discussed in the subsequent paragraphs:

Grid Scheduling: Broberg *et al.* have given detailed survey of economy based grid scheduling algorithm [1]. Major emphasis of these surveys is economy based grid scheduling environment. Recently, several works have been reported in this area. Xiao *et al.* have proposed an incentive-based scheduling algorithm for market based grid computing environment. This is a decentralized algorithm that uses a peer-to-peer (*P2P*) scheduling framework. They have developed the algorithm with dual objective for providing sufficient incentives to both the players (resource provider and consumers) in the market [26].

DELAY Strategy: The scheduling algorithms immediately dispatch the jobs upon their arrival. However, if a batch with large number of jobs is arrive in the full load condition both the existing and newly arrived processes suffer from poor response time. However, if the execution of newly arrived job is delayed until one of the currently running processes terminates, overall performance of the system will improve [9]. As per the literature survey, the DELAY strategy proposed by Hui and Chanson is not updated by any other researcher except the work reported by Zhang and Lu [27].

Content Aware Load Balancing: Another criterion for choosing a node is based on some special property of the consumer's processes or node called content aware scheduling. Casalicchio and Colajanni propose a new scheduling policy called Client-Aware Policy (CAP). CAP classifies the client requests on the basis of their expected impact on main server components [2]. Cherkasova and Karlsson have developed a Workload Aware Request Distribution Strategy (WARD) for clustered web servers. This strategy considers the workload properties and determines a set of most frequently used files and called them as core. These files are served by any server in the cluster while the remaining files are called as part files and partitioned between different servers in a manner such that each partition is served by different server in cluster [5].

Trust Management in Resource Management Algorithms: The large scale distributed computing environment involves a large number of resource providers and resource consumers. To properly distribute and execute the processes in such environments a lot of interaction is required. There is a requirement of some mechanism of building trust between these participants. Etalle *et al.* suggest that trust management policy must handle the issues related to the regret management and its accountability. They argued that opportunity must be given to a resource provider to rebuild the lost reputation. Regret management will give the opportunity for rebuilding the lost reputation [7].

Dynamic Content Handling Techniques: There exists a large body of research work on various caching techniques. However, the focus of this literature survey is database caching. *Memcached* is among the most popular object caching tool. This tool uses large hash tables for caching of the objects. Oracle's *Times Ten In-Memory* database is also another popular product providing faster access to the relational databases using the same programming interface.

Data Access and Integration from Disparate Sources: Literature in the field of data access and integration services contains some recent research projects including OGSA-DAI, SDO, Oracle Data Service Integrator and Microsoft SQL Azure Database Service. Comito *et al.* proposed a service based data integration architecture called Grid Data Integration System (GDIS) [6]. Gallo *et al.* have presented architecture for data access and integration specially designed for epidemic monitoring and simulation system based on grid integration services. It follows mediator based architecture as it is suitable for epidemic system [8].

Resource Discovery under Peer-to-Peer, Grid and Cloud: In various types of DCE, resources are diverse in nature e.g. special software, storage or network device, a specific processor, or some scientific instrument. Naseer and Stergioulas have classified these resources in two categories namely physical and logical. The physical resources are the basic hardware that creates the infrastructure. The logical resources are data set, knowledge and application resources [24]. Monitoring and Discovery System (MDS4) bundled with Globus toolkit is also a popular service.

4. APPROACH FOLLOWED

This research work makes several contributions towards the performance enhancement of the scheduling algorithms and resource management techniques used in various categories of DCEs.

1. Developed an economy-based Grid scheduling algorithm named as maximum utility (MU) algorithm. This algorithm maximizes the utility for both the resource providers and the consumers. The maximum utility for the resource provider indicates the higher profit and for the resource consumers, the maximum utility indicates successful execution of users processes with the lower cost of execution. The algorithm uses the market based parameters to determine the utility value for both the resource consumers and the resource

providers [16]. The incentive chosen by Xiao *et al*. are successful execution of the consumers' processes and the fairness in the market for resource providers. However, the proposed method uses the cost of execution of consumers' process and profit of providers' resources.

2. Modified the DELAY strategy proposed by Hui and Chanson and developed *mDelay* algorithm. If the distributed system is under the full load situation, the scheduling of new jobs is delayed instead of dispatching them to one of the overloaded workstations. The proposed *mDelay* algorithm produces better batch completion time and improved load balancing at workstations [13]. Hui and Chenson have taken six parameters to measure throughput and fairness of the algorithm. Instead of taking all these parameters for the comparison we have considered only two parameters to represents the fairness and throughput of the scheduling algorithms. The parameters are batch completion time and the workstation processing completion time.
3. Developed a new decentralized content aware load balancing strategy called workload and client aware policy (WCAP). The content-aware load balancing strategies use the services or content requested for the allotment of a process to a node. To incorporate content awareness existing load balancing policies used in DCEs have been modified [14], [20]. This policy reduces the search span in resource matching process.
4. Proposed and implemented a trust management and reliability policy for distributed scheduling algorithm used in the collaborative computing environment. Trust management is achieved with the concept of bidirectional reputation points that are assigned to resource providers and resource consumers. Moreover, concept of reliability is also used to ensure failsafe execution of the processes. This policy also uses the activeness, ratio of positive to negative reputation points obtained by participants and recently earned reputation points along the reputation points [18].
5. Designed and developed a Distributed Hash-Based Database Caching (DHBDC) technique for web services and multitier web application deployed over DCEs. This database caching is implemented with centralized cache control for better performance. It uses larger distributed hash tables stored in the memory of the servers in cache cluster [23], [11].
6. Developed a web service based API for data access and integration called Grid Data Access and Integration Service (GDAIS). This service provides facility to access and manipulate data from various data stores. The API is simple, generic in terms of databases as well as the implementation architecture. It provides simple interface with less resource consumption for the users of data stores [17].
7. Developed the web service based approach for resource registration and discovery (RRD) in distributed computing environment. The service can be used in cluster computing, P2P computing, grid computing and cloud computing environment. This web service also provides facility to discover the suitable resources, matching it with the processes and booking of the most appropriate resource. The proposed service is a two-level decentralized and hierarchical method to avoid single point failure and better performance [21].
8. Presented the distributed computing solutions to one scientific and one commercial application viz. e-Learning and data mining [15], [19]. Suggested appropriate architectural solutions to e-Learning applications of various size based on data size and number of users. Also proposed the concept of Knowledge Discovery from Data as a Service (KDDaaS) to develop Knowledge Cloud.

5. METHODOLOGY

To test the proposed algorithm a custom simulation environment is developed called *EcoGrid* [12], [22]. *Globus Toolkit* is also used to develop grid services. The development and evaluation of scheduling algorithms and strategies are performed on these two tools. *Globus Toolkit* is a set of tools and protocols that enables users to create real *Grid*. Globus contains five categories of components viz. common runtime, security, data management, information services and execution management. The common runtime component provides the libraries needed to build the services. The communication among various computing elements is secured by the security component. The data management component facilitates management of huge data set in VO. The information service is responsible for monitoring and discovery of the resources in VO. The execution management deals with the various activities related to job's execution e.g. initialization, monitoring, scheduling etc.

The scheduling algorithms developed for *Grid* viz. modified delay strategy and content aware load balancing needed a specific test bed. Several test-beds like *GridSim* [25] and *SimGrid* [3] are already available but these simulation environments do not support some necessary features. To satisfy the specific needs *EcoGrid* is developed as a test-bed for grid scheduling algorithms based on dynamic load balancing. *EcoGrid* is dynamically configurable and optimizes the cost of execution of a process and maximizes the profit of a service provider system. As compared to other commonly used grid simulation tools like *GridSim* and *SimGrid*, *EcoGrid* provides more enhanced features like configurability, scalability and extensibility. This object-oriented simulator also supports execution of Java applications as well as simulation of both the economy based and non-economy based scheduling algorithms. The proposed algorithms were evaluated with the other state of the art algorithms proposed in the literature using *EcoGrid*. The comparisons indicate a great deal of improvements. EcoGrid represents the resources in the form of objects rather than the threads, resulting in the low memory use and high scalability as observed from Figure 1.

The proposed strategies are tested on synthetic and real workloads. Synthetic workload is generated using appropriate statistical distribution. The values for each of the parameters are generated using the most appropriate statistical distribution for that parameter. For example, since the process arrival time follows Poisson distribution, the arrival time is generated using Poisson distribution. The real workload traces are taken from two sources. The first source is Grid Workload Archive (GWA). Iosup *et al.* created a standard Grid Workload Format (GWF) [10] that is followed by the logs in GWA.

The second source of real workload trace is the parallel workload archive maintained by Feitelson. These workload traces are collected from various large scale parallel systems from various places around the world. The traces are already converted in Standard Workload Format (SWF) to follow a uniform formatting [4]. The GWF and SWF contain several fields namely, job-id, submit time, Waiting Time, execution time, number of allocated processors, amount of used memory, requested number of processors, requested time, requested memory, status, user-id,

group-id, queue number, originating site-id, last run site-id, structures containing the preceding job-id and the think time between preceding and current job, used network bandwidth, used local disk, used resource, requested platform (CPU, OS and Version), project-id, VO-id etc. However, few attributed were synthetically added to the workload traces as they do not posses several attributes needed in economy based scheduling. The open source software were preferred for implementations related to this research as they are available quite easily and have wide user base.

6. RESULTS

This section presents main results of our PhD research work, for brevity other results are skipped can be found in our papers. According to the requirement of the algorithms, for each of the experiments a different experimental setup is created. *EcoGrid* provide enhanced features and improves performance obtained from the existing well established simulators like the *GridSim* and the *SimGrid*. *EcoGrid* performs better in terms of memory usage, number of processes, number of resources and the runtime performance as compared to GridSim that can be concluded from Figure 1 and Figure 2.

Similarly the *mDelay* strategy, developed by modifying DELAY strategy proposed by Hui and Chanson [9], produces better results than DELAY strategy. The results are given in Figure 3 and Figure 4. This experiment is performed to test algorithm on a test-bed having of hundred workstations. The size of normal batches is eighty and the size of delayed batches is three hundred. Six different types of processors are considered.

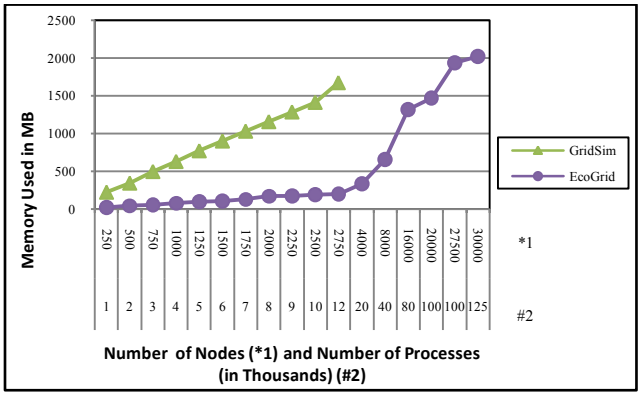

Figure 1. Memory Usage of *EcoGrid* and the *GridSim*

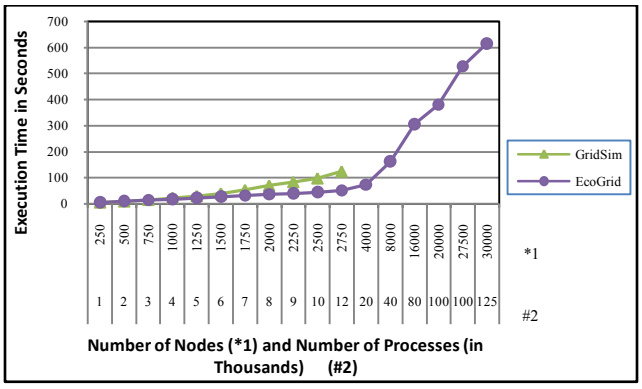

Figure 2. Execution Time of *EcoGrid* and the *GridSim*

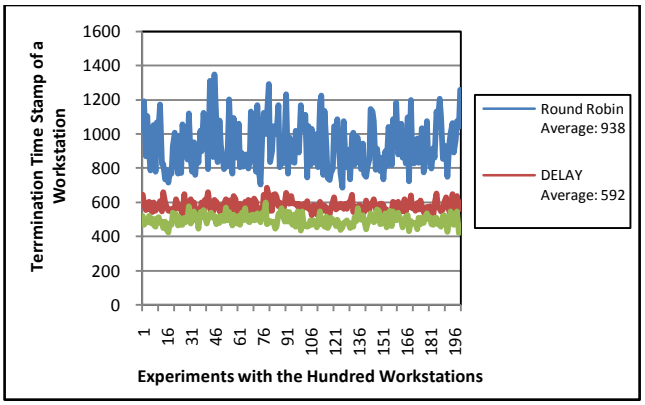

Figure 3. Average Completion times of Processes of all Workstations using Distributed Round Robin, *DELAY* and *mDelay*

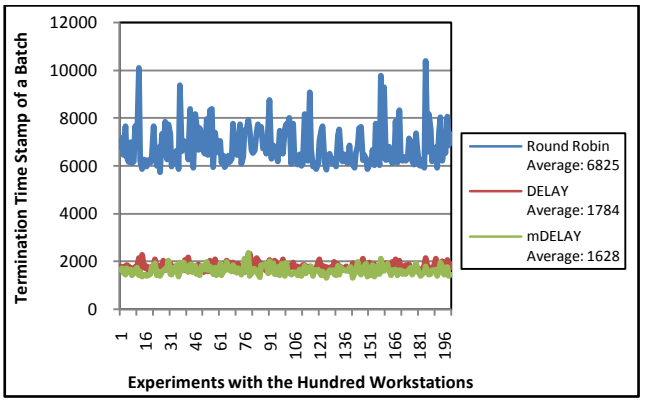

Figure 4. Average Completion Time of Various Batches using Distributed Round Robin, *DELAY* and *mDelay*

In order to test the effect of execution time of jobs, one large process is created in each of the normal batches and seven large processes are created in each of the delayed batches arriving at system. In this experiment the data is collected from the two hundred iterations of the experiment. In one repetition average of completion time of last candidate process at each of the workstations is calculated. At the same time, the average of the completion time of the last candidate process from each of the batches is also calculated. The *mDelay* produces 16% improvement in average completion time of last process at each workstation and 10 % improvement in average completion time of each batch.

7. CONCLUSION AND FUTURE WORK

The improvement in DCEs will be an important step towards the improvement in the performance of applications (including web applications and web services) deployed over such DCEs. This research is having three major contributions. The first is in the area of distributed scheduling or resource management algorithms, where, four distributed scheduling algorithms MU, *mDelay*, WCAP and Trust Management and Reliability Policy have been developed. These algorithms may also be adopted in any of the existing distributed schedulers. The second type of strategies is service based approach to data access & integration and resource registration & discovery applicable in large distributed computing environment. The third and final technique is a database caching scheme developed for the multitier web

applications deployed on the *Cluster*, *Grid* or *Cloud*. This strategy has wide applicability in web applications since it improves the overall performance of the application. We feel that these algorithms shall improve the overall performance of the existing DCEs.

No work is enough to directly apply real world scenario, several enhancements are needed to apply these problems to real world applications. We proposed architectural solution to e-Learning and data mining applications, their implementation is an important future extension of this work.

8. REFERENCES

[1] J. Broberg, S. Venugopal, R. Buyya, "Market-oriented grids and utility computing: the state-of-the-art and future directions," *Journal of Computing,* Springer Verlag, Germany Vol. 6, Issue 3, 2008, pp. 255-276.

[2] E. Casalicchio, M. Colajanni, "A client-aware dispatching algorithm for web clusters providing multiple services," 10th International World Wide Web Conference, May 2001, pp. 535-544.

[3] H. Casanova, "*SimGrid*: a toolkit for the simulation of application scheduling," *IEEE International Symposium on Cluster Computing and the Grid*, May 2001, pp. 430-437.

[4] S.J. Chapin, W. Cirne, D.G. Feitelson, J.P. Jones, S.T. Leutenegger, U. Schwiegelshohn, W. Smith, D. Talby, "Benchmarks and standards for the evaluation of parallel job schedulers," *Job Scheduling Strategies for Parallel Processing*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1659, 1999, pp. 66-89.

[5] L. Cherkasova, M. Karlsson, "Scalable web server cluster design with workload-aware request distribution strategy," *3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, San Jose, CA, Jun 2001, pp. 212–221.

[6] C. Comito, D. Talia, T. Paolo, "Grid services: principles, implementations and use," *International Journal of Web and Grid Services*, Inderscience Publishers, 2005, pp. 48-68.

[7] S. Etalle, J.D. Hartog, S. Marsh, "Trust and punishment," *1st International Conference on Autonomic Computing and Communication Systems,* October 2007.

[8] E. Gallo, H. F. Gagliardi, F. A. B. D. Silva, V. C. Neto, D. Alves, "GISE-a data access and integration service of epidemiological data for a grid-based monitoring and simulation system," *40th Annual Simulation Symposium*, Mar 2007, pp. 267-274.

[9] C.C. Hui, S.T. Chanson, "Improved strategies for dynamic load balancing," *IEEE Concurrency*, Vol. 7, July 1999.

[10] A. Iosup, H. Li, C. Dumitrescu, L. Wolters, D.H.J.E Pema, "The grid workload format," *http://gwa.ewi.tudelft.nl/ TheGridWorkloadFormat_v001.pdf*, Nov 2006.

[11] H. Mehta, P. Kanungo, M. Chandwani, "Performance enhancement of scheduling algorithms in web server clusters using improved dynamic load balancing policies," *2nd National Conference, INDIACom-2008 Computing For Nation Development*, New Delhi, Feb 2008, pp. 651-656.

[12] H. Mehta, P. Kanungo and M. Chandwani, "Performance evaluation of grid simulators using profilers," *2nd International Conference on Computer and Automation Engineering*, Singapore Vol. 2, February 2010, pp. 74-77.

[13] H. Mehta, P. Kanungo, M. Chandwani, "A modified delay strategy for dynamic load balancing in cluster and grid environment," *International Conference on Information Science and Applications (ICISA-2010)*, Seoul, Korea, Apr 2010, pp. 1-8.

[14] H. Mehta, P. Kanungo, M. Chandwani, "A content aware load balancing algorithm for grid computing environment," *3rd CSI National Conference on Education & Research*, Mar 2010, pp. 95-101.

[15] H. Mehta, P. Kanungo and M. Chandwani "Towards development of a distributed e-Learning ecosystem*2nd International Conference on Technology for Education*, IIT Mumbai, July 2010, pp. 68-72.

[16] H. Mehta, P. Kanungo, M. Chandwani, "Maximum utility meta-scheduling algorithm for users of economy based grid scheduling environment," *3rd International Conference on Contemporary Computing*, Aug 2010, pp. 23-33.

[17] H. Mehta, P. Kanungo, M. Chandwani, "Generic data access and integration service under distributed computing environment," *International Journal of Grid Computing and Applications*, Vol. 1, No. 1, Sep 2010, pp. 14-21.

[18] H. Mehta, P. Kanungo, M. Chandwani, "Trust managing maximum utility meta-scheduling algorithm for economy based grid scheduling environment," Accepted in *2nd International Conference on Advanced Computing*, Chennai, Dec 2010.

[19] H. Mehta, P. Kanungo and M. Chandwani "An architectural roadmap of building knowledge cloud," Communicated.

[20] H. Mehta, P. Kanungo, M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environment," Accepted in *International Conference and Workshop on Emerging Trends and Technology*, Mumbai, Feb 2011.

[21] H. Mehta, P. Kanungo, M. Chandwani, "A service based approach of resource registration and discovery in distributed computing environment," Communicated.

[22] H. Mehta, P. Kanungo, M. Chandwani, "*EcoGrid*: a dynamically configurable object oriented simulation environment for economy-based grid scheduling algorithms," Communicated.

[23] H. Mehta, P. Kanungo and M. Chandwani, "Distributed database caching for multitier web applications," Accepted in *International Conference and Workshop on Emerging Trends and Technology*, Mumbai, Feb 2011.

[24] A. Naseer, I.K. Stergioulas, "Resource discovery in grids and other distributed environments: states of the art," *Multiagent and Grid Systems - An International Journal*, IOS Press, 2006.

[25] A. Sulistio, U. Cibej, S. Venugopal, B. Robic, R. Buyya, "A toolkit for modeling and simulating data grids: an extension to *GridSim*," *Concurrency and Computation: Practice and Experience*, Wiley Press, Jan 2008, pp. 1591-1609.

[26] L. Xiao, Y. Zhu, L.M. Ni, Z. Xu, "Incentive-based scheduling for market-like computational grids," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 7, Jul 2008, pp. 903-913.

[27] J. Zhang, X. Lu, "A dynamic job scheduling algorithm for computational grid," *2nd International Workshop Grid and Cooperative Computing*, Dec 2003, pp. 40-47.