

A Multi-agent System-Based Resource Advertisement Model for Grid Environments

¹Muntasir Al-Asfoor, ¹Maria Fasli and ¹Salah Al-Majeed

¹School of Computer Science and Electronic Engineering, Essex University, Wivenhoe Park, Colchester, UK, CO4 3SQ
{mjalas, mfasli, ssaleha}@essex.ac.uk

Abstract. The field of resource sharing has become an increasingly important computing field of study for both industry and academia. Facilitating the use of geographically distributed resources is a complex task encompassing many sub tasks. The most important sub-tasks are advertising the available resources across the network and the discovery of these resources. In this paper, the task of advertising the resources has been studied from a network point of view. An advertisement model has been designed using multi-agent system techniques to achieve decentralization and to cope with the dynamic nature and behaviour of the system. The design has been implemented and tested under a simulation environment using the network simulator ns-2. The system evaluation has shown promising results in terms of the system performance itself and the effect of the advertising overhead on a working network's performance.

Keywords: Grid, Multi-Agent Systems, Resource Advertisement, Network Performance.

1 Introduction

In the early-to-mid 1990s, there were numerous research projects underway in the academic and industry communities that were focused on distributed computing. One key area of research focus was on developing tools that would allow distributed high performance computing systems to act like one large computer [1]. Analogous to an electricity power grid, Grid computing views computing, storage, data sets, expensive scientific instruments and so on as utilities to be delivered over the Internet seamlessly, transparently and dynamically as and when needed, by the virtualization of these resources [9]. A Grid is a very large scale, generalised distributed system that can scale to Internet-size environments with machines distributed across multiple organisations and administrative domains [3]. There are two issues to be addressed by any Grid resources management system: firstly, the advertisement (dissemination) of the resources information across the Grid nodes, that is how nodes should describe their resources and when to advertise them. . For instance, if the resource is a processing unit some nodes might describe it as "processor" while the other as "CPU"; the two term are syntactically different but semantically the same. Furthermore, as the advertising process is a message passing activity, it is important to consider the effect of this activity on the network performance. Secondly, the other

fundamental problem in Grid computing is the discovery of resources in such a heterogeneous computing environment as well as dynamicity of resource enrolment and leaving process. Resource discovery is a key concept in a distributed Grid environment; it defines a process for locating services providers and retrieving service descriptions [11].

The Grid and Multi-agent Systems communities have both developed concepts and mechanisms for open distributed systems [7]. Agents with different expertise and computational resources which may be distributed over a network can come together and coordinate to solve a complex task [6].

In this research, we have focused on the advertisement process from a networking point of view. Furthermore, we have studied the effect of this process on the network performance in terms of time Delay required to deliver the advertisements. An advertisement model has been designed based on multi-agent system techniques; furthermore, simulation tools have been used to implement and test the design.

The rest of the paper is organised as follows: section 2 discusses related work while section 3 is devoted to a detailed explanation of the proposed system architecture. Section 4 shows the system evaluation and the experimental results of the test that been conducted in a simulated environment. The paper ends with the conclusions and a description of future work.

2 Related Work

Researchers in the industry and academia have developed many frameworks and tools for managing the resources description and discovery in Grid environments. The Globus toolkit [10] Monitoring and Discovery System (MDS4) includes a set of monitoring and discovery techniques for services and resources in a distributed environment. It makes use of WSRF (Web Services Resource Framework) but in a centralised way. With this centralised way for managing the monitoring and discovery of resources, any problem that might occur in the node where all the indexing information is stored, will lead to the halt of the entire system. A further weakness of Globus is the use of static periodic updating which might result in unnecessary communications. In addition to that, the Globus update strategy cannot adapt itself as a response to system load but depends on manual intervention.

Another Grid system is Condor [12], which is a specialized workload management system for computing-intensive jobs. It provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Condor has a centralised job scheduler which accepts the jobs which are submitted from the condor workstations. One of the weaknesses of Condor is its centralised scheduler which could be a single point of failure. Furthermore, Condor uses a push update strategy which causes interventions to the system to change the resources status.

Grid resource brokering algorithms are presented in [5] for enabling advance reservations and resource selection based on performance predictions. They suggested a resource broker that selects resources by using the actual job requirements provided by the job owners. The main drawback of this method is how to determine the job

requirement which needs an intervention from the user to fix the job requirements by providing an estimated amount of resources required to run the job..

A Grid environment was studied in [4]. They have utilized semantic matching in their work with the aim of making the Grid system more transparent in the application level. In this research the authors tried to achieve some sort of scalability by imposing homogeneity of resources on a virtual organisation. Imposing homogeneity is not widely accepted as the main characteristic of Grid systems is to be heterogeneous and each node could have different hardware or software infrastructure.

Based on semantic web technologies, a resource discovery middleware has been proposed by [8]. In this work, they employed the semantic web discovery technologies in the field of Grid resource discovery by adding machine-process-able explicit knowledge into the interaction between pervasive devices and Grid services. The system uses a profile model to describe Grid resources. According to their results, the system has showed an increase in service discovery time compared to the traditional service discovery mechanism while a significant improvement in the system function has been obtained. The use of profile model to describe the resources will lead to the use of statistical matching techniques which does not take the semantic features of the resources concepts into account. It could be part of the matching process but not the only way to determine the degree of similarity among concepts that describe the resources.

We have concluded that most of the work that has been undertaken in the field of Grid resource management has ignored or paid less attention to some important issues. The first issue to be considered is the performance of the advertisement part of the system and its effect on the overall system efficiency. Improving the network performance has a big role to play on the overall Grid management systems performance. As a Grid is a huge network system, the main factor that effects the quality of service (QoS) of that system is the amount of time required to advertise the provided resources and discover these resources. Monitoring the message passing and decreasing the communication overheads would increase the system performance and lead to improving the average system delay and increasing the network throughput and accordingly improve the overall QoS of the Grid system. In this research we have focused on these issues by designing a multi agent Grid system architecture which deals with the dynamicity and heterogeneity behavior of the Grid. The system does not depend on traditional frameworks but works as a flat decentralized network.

3 Proposed System Architecture

3.1 The Basic System Components

Technically, a Grid resource management system involves three major aspects: Registration, Advertisement and Location.

In the system design, it is assumed that any node joins the system network is considered as a registered node and has the right to advertise its available resources and request the needed resources to accomplish its applications requirements. The new registered node must have the required agents to be compatible with the system. Each agent is responsible for a specific job(s).

The system consists of N number of nodes each of which has many functions which are shown in Fig. 1. To conform with the principle of decentralisation and taking into account heterogeneity, each node has its own agents and each agent has specific functionality and could adapt itself to cope with the node specification (heterogeneity). Furthermore, each node has its own local cache to store its available resources information.

There are five types of agents in the system classified according to their functionality as follows:

- Database Management Agent (**DBMA**): responsible for all databases related functions (Insert, Delete, Update, and Search).
- Resource Collection and Description Agent (**RCDA**): responsible for collecting the available local resources and describing them ontologically.
- Matchmaking Agent (**MA**): responsible for matching requests queries with the local resources information in the node's local cache.
- Network Agent (**NA**): responsible for all the network related issues (sending description messages, receiving acknowledge and monitoring network performance).
- Resources' Broker (**RB**): responsible for negotiating among nodes over the available resources.

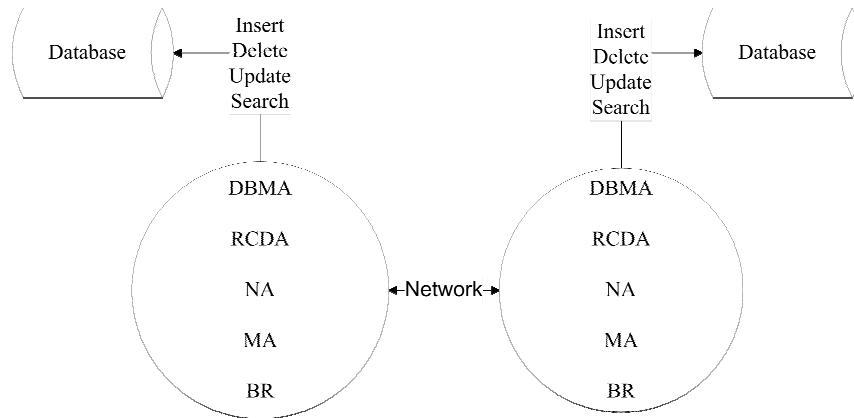


Fig. 1 The basic components in a typical node

3.2 The Advertisement Process

Any node that joins the system for the first time must broadcast its available resources information to all its neighbours¹. Before advertising the resources information, the node must have the required agents locally as part of the registration process.

After finishing the registration process, the new node should start advertising its resources' information to its neighbours. The process of advertisement consists of three steps which are handled by three types of agents as shown in Fig. 2.

¹ Neighbors of the node are all the nodes that have a direct connection to it.

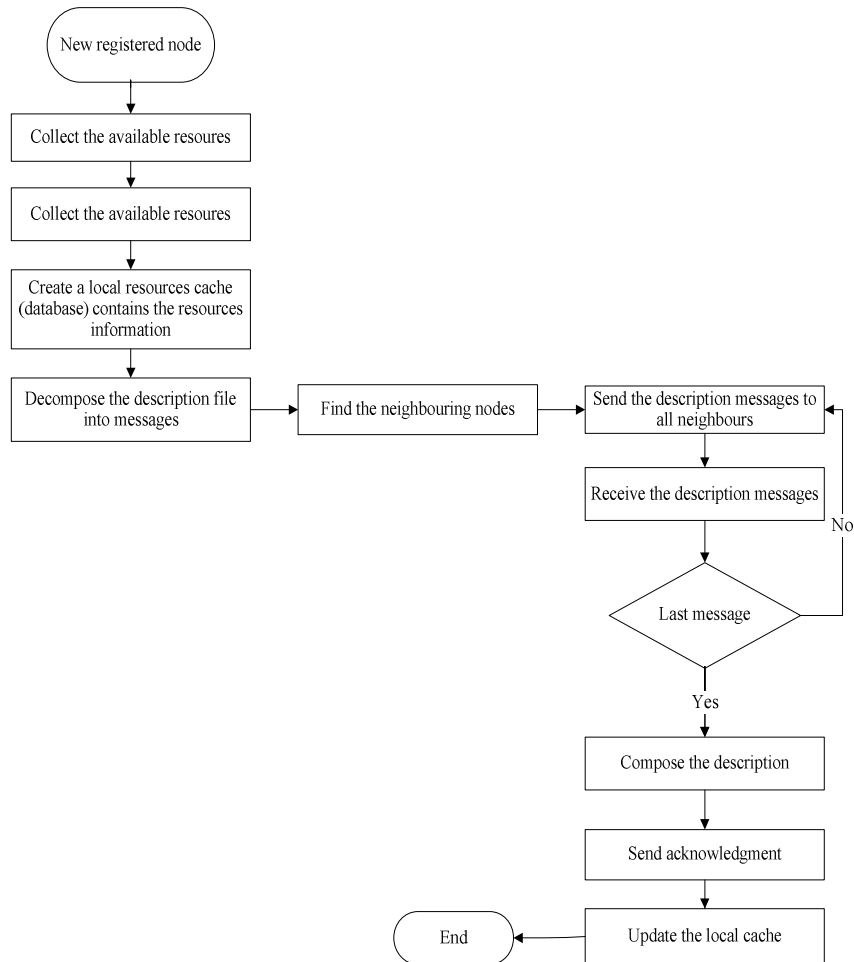


Fig. 2 The Advertisement model

The first step is to collect information about the available resources and describe them ontologically and the RCDA is responsible for this task. Since in principle, nodes are highly heterogeneous, RCDA can adapt itself according to the node's specifications, for instance, it must use the ontology description that the node uses. The resources to be described are just the ones that can be shared with the other nodes in the system (i.e. they are not needed by the owner node for the moment. The RCDA creates the ontology description by connecting to a standard ontology database like WordNet² according to the node specification.

Having all the resources information stored in one place (node) makes the system less fault-tolerant. The node which holds the resources information could become a single point of failure, if any problem arises with this node, which could

² See <http://wordnet.princeton.edu/>

lead to overall system collapse. For this reason, and to achieve system extensibility and cope with the dynamic nature of the environment, we propose a fully distributed and decentralised database to store the resources information within our proposed system.

The DBMA is responsible for creating and managing the node's local caches; after the node registration takes place, the DBMA creates an empty local database which holds the node's resources description. Furthermore, the DBMA provides the basic database functions like (data insertion, update, alteration, etc.). Another function of the DBMA is to inform the NA of any changes to the database in order to send an update message to the neighbouring nodes. The NA plays a crucial role during the advertisement process. As shown in section 2 there have been many works in the field of Grid resources management, few of them have been concerned with the network performance during the advertisement process.

As the main part of the advertisement process is a network related process, we have focused more on network performance as an evaluation measure of the advertisement; the best advertisement method is the one with the best network performance. The advertisement is the process of sending the resources' description to the nodes in the Grid system. Sending the resources description is a message passing activity; balancing the number of messages and associated traffic, in order to reduce the communications' overhead is an important factor which affects the overall system performance.

On the sender side, and as a result of the description process, the RCDA passes the description file to the NA; the size of the description file varies according to the amount of resources available and the way the RCDA describes them. For network purposes, the NA should divide the description file into messages with each message containing part of the resource description information. The division depends on the message size; there is a trade-off between message size and the required time to deliver the description information. Smaller message size means a faster delivery time but at the same time increase in the number of messages and vice versa. After that, the NA should start to find the neighbours with direct connections to the sender node. For example, in Fig. 3, the neighbours of the node number 1 are nodes (2,3 and 5); This organisational model will form some sort of local community with each node having some knowledge about its neighbours that could facilitates the resource discovery afterwards. Accordingly, the NA should start sending the description messages to the neighbours.

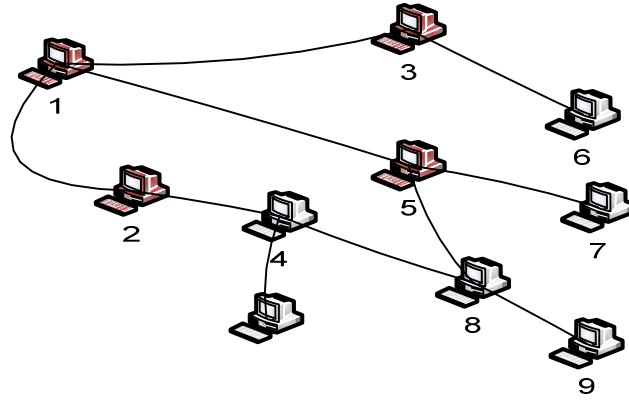


Fig. 3 The Nodes Neighborhood

On the receiver side the NA receives the description messages one by one and checks the message id and when receiving the last message it starts to compose the description file. The receiver NA should send an acknowledgment message to the sender NA to confirm that the description information has been received in its entirety.

After receiving the description, the NA passes the description file to the DBMA to create a new entry for the new node's resource information in the local database.

3.3 Resources' Availability Update

As the Grid system is dynamic and due to resources' use, resources availability could change over time. Accordingly, to keep the nodes up to date with these changes, the system should keep updating the nodes local caches with all the changes that occur over time.

There are two updating strategies; the first strategy is to update the system periodically by disseminating the resources' information through the system each time slice [references?]. This strategy does not take into account the system workload or nodes' needs but updates all the nodes blindly. The second strategy is to update the system due to some event (event-driven update) [references]. Using this strategy the nodes should update their resources caches during the event of a request to make sure that the discovery process takes place according to the latest update.

Both updating strategies could make unnecessary updates which affect the overall system performance. For the first strategy, the resources information might remain the same so the update will cause extra and unnecessary advertisement messages to be sent which means more communications across the system. For the second strategy, the update takes place during the resources request time without taking in mind that the request would not be satisfied by the requested node due to lack of resources. In this case, an unnecessary update which causes extra advertisement messages will result in communications overhead across the system.

[13], have presented a new technique for updating the system by using an expiry time stamp called UST (Updatable Service Lifetime). The expiry time stamp is computed by taking into account both strategies to ensure that all the nodes' caches will be updated each time slice and during the event of resources request. They have

used a tree structure to represent the agents which represent the nodes through the system. UST computation encompasses the use of three factors: the frequency of changes in the agent's status, the relation between the number of agent's children to the overall number of agents in the system, and the list of active children which frequently contacted their parent.

Based on their strategy we have adapted a time stamp computation technique in order to apply it to our network as follows: the time stamp computation will depend on the history of change for the node and the number of neighbouring nodes. Logically, it is more likely for active nodes³ to be changed in the future. So based on historical changes the expected change time is calculated by the following forecasting formula:

$$RICT = a + xb \quad (1)$$

Where $a = y' - bx'$ and $b = \frac{\Sigma(x-x')(y-y')}{(x-x')^2}$, $RCIT$: is the resource information change time

Where:

x : is the count number of changes for the node.

y : is the time at which the change has occurred.

x' and y' are the average values of x and y .

After calculating the historical effect of the number of changes on the next update time, the next step is to calculate the effect of the number of neighbours on the calculation of the expiry time stamp using the following equation:

$$NCF = \frac{NoC}{ANoC} \quad (2)$$

Where,

NCF : is the number of connections factor.

NoC : is the number of connections (neighbours) of the node.

$ANoC$: is the average number of connections for each node.

At the end, the calculation of the expiry time stamp is done by taking into consideration both equations as follows:

$$ETS = RICT(1 + NCF) \quad (3)$$

Where:

ETS : is the expiry time stamp.

4. System Evaluation and Experimental Results

To evaluate the proposed system performance we have engineered a number of tests using the well-known network simulator ns-2 (v. 2.29)⁴; a trace file was input to the well-known network simulator ns-2 and traffic time recorded in the output. The output serves to calculate the time delay.

First of all, the average amount of time required for the advertisement messages has been calculated. Furthermore, since we would like the system to be expandable, the test has addressed the effects of increasing the number of nodes across the system. Fig. 4 shows the delay time in ms for an advertisement message to arrive at its final

³ Active nodes are the ones that their resources information is changing more frequently.

⁴ <http://www.isi.edu/nsnam/ns/>

destination. The results show a reasonable average delay time is 0.021576 ms with a standard deviation equal to 0.007095.

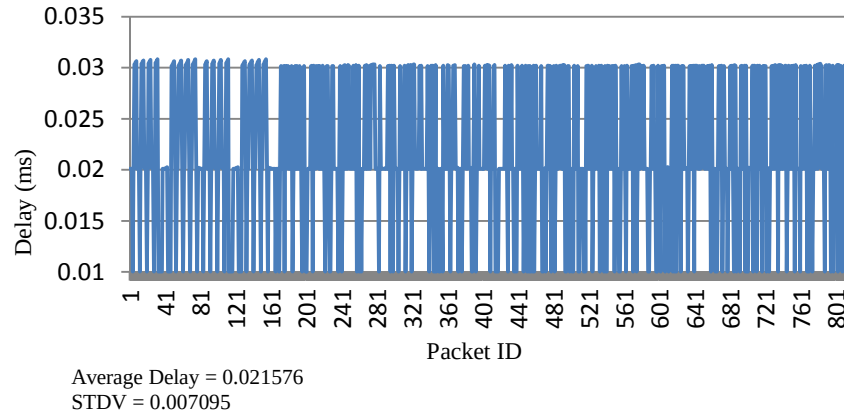


Fig. 4 packet delay for advertisement messages alone

To assess the system's ability to cope with expandability, the second test has evaluated the effect of increasing the number of nodes on system performance. As the number of nodes increases the number of messages to be sent across the system increases since more advertisements will be sent. As shown in Fig. 5, the delay time has increased dramatically in response to the increase in the number of nodes. The increase in the delay is logical since more advertisements will be sent through the system but the average delay time still remains reasonable under 0.05 ms.

In the third test, we have evaluated the effects of our advertisement process in an over-utilized network. We have designed a normal network with UDP⁵ agents that send and receive data packets in addition to our system advertisement messages. The idea behind this test is to evaluate the network performance with our advertisements making use of the network resources with CBR⁶ traffic already in use. Fig. 5 shows the average delay time with and without advertisements. The test has showed that there is a small (hardly noticeable) increase in the delay time when the advertisements have been incorporated into the system. The small increase in the delay time has occurred because the advertisement messages have made use (shared) the network resources alongside the other network activities.

⁵ UDP stands for User Datagram Protocol.

⁶ CBR stands for Constant Bit Rate

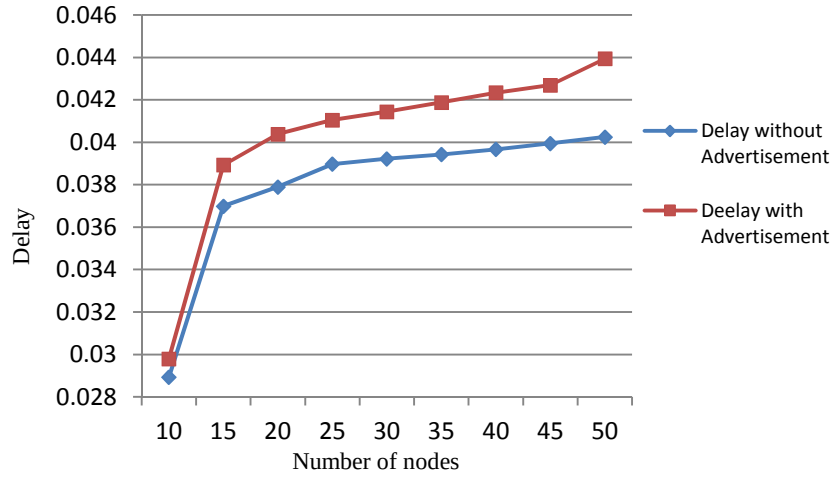


Fig. 5 Packet delay for CBR with and without advertisements

The difference in delay is small and has no effect on system performance because of the following reasons. First of all the message size; in the system architecture the message has been designed to be small in size so that it does not need large network bandwidth to be sent. Instead, the message could make use of any available small bandwidth which is more probably not used by other network activity. The second reason is the advertisement system architecture itself which sends advertisement messages just to the neighbors' nodes that have direct connections which avoids wasting time in routing.

5. Conclusion and Future Work

The rapid development in the scientific research and business requirements for computational and data resources have led to the need for efficient ways to manage the process of describing and discovering these resources. We have studied the advertisement part of the resource management system from a network point of view. Firstly, we have proposed a multi-agent system architecture for resource advertising which increases interoperability, copes with nodes heterogeneity and supports system scalability. Secondly, we have introduced a new mechanism for constructing and subsequently disseminating resource advertisements to the neighboring nodes with the aim of decreasing the network overhead by dealing with subsets of nodes rather than the whole network. Finally, we have proposed a modified method to update resource information which is more efficient in terms of network's communications and overcome the unnecessary update messages. Furthermore, the system lacks a single point of control which makes it more immune to faults that might occur to any node across the network. In terms of performance, the small message size has shown a good result in terms of network performance by making use of a small network bandwidth which does not affect the work of the other network activities that are taking place while the advertisement system works.

There are several avenues for future work. Firstly, the system could be expanded to perform more resource management jobs. More specifically, a request processing part could be added to the system to monitor the number of requests being

sent through the system in order to improve the system performance by grouping the nodes across the network based on its functionality as well as geographic locations. Grouping the nodes gives the system some sort of knowledge about the nodes' resources and decreases the number of requests which in turn improves the network performance. More after to cope with nodes heterogeneity it is very important to provide some sort of semantic matching in order to discover resources based on their properties rather than simple keyword matching.

References

1. Abbas, A. (2004). Grid Computing: a Practical Guide to Technology and Applications. Charles River Media.
2. Amarnath, B., Somasundaram, T., Ellappan, M. And Buyya R.(2009).Ontology-based Grid Resource Management. Software, Practice and Experience (39), pp. 1419-1438.
3. Bote-Lorenzo, M., Yannis, A., and G´omez-S´anchez, E.(2004). Grid Characteristics and Uses: A Grid Definition Across Grids, Lecture Notes in Computer Science 2970, pp. 291–298.
4. Brooke, J., Fellows, D., Garwood, K., and Goble, C. (2004). Semantic matching of Grid Resource Descriptions. In 2nd European Across-Grids Conference. Lecture Notes in Computer Science 3165, pp.240-249.
5. Elmroth, E., and Tordsson, J. (2008). Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions. Future Generation Computer Systems 24(2008): 585–593.
6. Fasli, M. (2007). Agent Technology for e-Commerce. John Wiley & sons Inc.
7. Fattahi, S. and Charkari, N. (2009). Distributed ACS Algorithm for Resource Discovery in Grid. International Conference on IT to Celebrate S. Charmonman's 72nd Birthday, March 2009, Thailand.
8. Guan, T., Zaluska, E. and De Roure, D.(2009). An Autonomic Service Discovery Mechanism to Support Pervasive Devices Accessing the Semantic Grid, International Journal of Autonomic Computing, pp. 34-49.
9. Ludwig, S. and Van Santen, P. (2002). A Grid Service Discovery Matchmaker Based on Ontology Description. In proceedings of the Second International EuroWeb2002 Conference, Oxford, UK, pp. 17-18.
10. Schopf, J., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M., and Chervenak, A.(2006). Monitoring the Grid with the Globus Toolkit MDS4. Journal of Physics: Conference Series 46(2006), pp. 521–525.
11. Timm, I. and Pawlaszczyk, D. (2005).Large Scale Multi-agent Simulation on the Grid. In Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) – (1), pp. 334-341.
12. Wright, D., Miller, K., and Livny, M. (Eds.) (2002). Condor - A Distributed Job Scheduler. In T. Tannenbaum, Beowulf Cluster Computing with Linux. The MIT Press.
13. Youssif, A. ,Ghalwash, A. and Ezz El Dien, M.(2010). An Efficient Decentralized Grid Service Advertisement Approach Using Multi-Agent System. Journal of Computer and Information Science 3(2010).