A Novel Decentralized Fuzzy Based Approach for Grid Job

Zeinab Khalilian[1], Seyed Javad Mirabedini[2]
[1]*Department Of Computer, Science And Research Branch, Islamic Azad University, Arak , Iran*
[2]*Engineering Department, Science And Research Branch, Islamic Azad University, Tehran, Iran*
zeinab.khalilian2011@gmail.com

***Abstract*—In this paper with the aid of fuzzy theory we present a new method for scheduling on Grid system. Grid computing is a technology to meet the growing computational requires. In fact grid computing is one of the most popular types of distributed system. Its aim is to produce an enormous, autonomous and effective virtual machine, and it is produced by collecting different nodes with the aim of sharing their data and computational power. This paper follows the identification of grid scheduling with the help of fuzzy theory and seeking to present a new method for grid scheduling with respect to exiting obstacles. In our method we use the intermediate load of nodes of each clusters, the average of computing power which determines the node premiership and job premiership as the input parameters of fuzzy system, and regarding to the output value of fuzzy system the suitable nodes determines. We evaluate the performance of our method with some grid scheduling methods. The results of the experiments show the efficiency of the proposed method in term of makespan and Standard deviation of the load of clusters.**

***Index Terms*—Grid, Decentralized Job scheduling, Fuzzy theory, P2P overlay.**

I. Introduction

Grid is a connected couple distributed system. It is created by some computing resources which are connected to find a solution for large-scale computational problems [1]. A computational grid is a huge, heterogeneous amount of independent machines, geographically distributed by interconnected networks [2]. The computational job sharing is a main application of grids. Grid consists of some resources which are dynamic and diverse and can be joined to the network and leave whenever the owner permits them for leaving or joining the group [3]. Grid resource management consists of four steps; resource discovery, resource selection, job scheduling and submission and monitoring of jobs [4]. The major factor for an effective grid resource management is the grid resources qualities[5]. Strategies for job scheduling and its algorithms are on the basis of considering the current system status[6].

In this paper with the help of fuzzy theory we propose a novel distributed job scheduling approach for grid systems. The focus of our approach is on scheduling phase of resource management. In our scheme, fuzzy theory is used by 3 parameters to select the most sufficient node for scheduling; the power of nodes processors which determines the computing power, the length of current job running on the nodes and the new entered job premiership.

Fuzzy theory is much less strict than the computation computers typically perform. Fuzzy theory offers various unique characterizes that make it a particularly good optional for many control problems. It is fundamentally strong since it does not need exact inputs[7]. Fuzzy theory manages the examination of knowledge with the aid of fuzzy sets. Tease sets can be shown a linguistic expression such as “Low”, “Medium, and “High” etc.[8].

Job scheduling is one of the most major issue for obtaining high performance on Grid systems. Several job scheduling strategies have already been proposed and carried out in different types of Grid systems. Lots of these strategies improve the job scheduling based on the change of the environments. In spite of the fact that many proposed job scheduling algorithms proved they are appropriate for a dynamic environment, only a few scheduling strategies have been proposed considers the current status of grid resources and the change of the environments.

The rest of this paper is as follows: in the next section we provide an overview of related works about grid job scheduling. In this section, we take a brief look at some works which are about grid job scheduling. The proposed approach is presented in section III. The performance evaluation is presented in Section IV; this section shows the results. Finally, in section V, we make a conclusion.

II. Literature Review

Job scheduling models are divided into three types [9][10]; Centralized model, hierarchical schemes and distributed schemes. In this section we present a brief overview of some works on grid job scheduling.

In the most of the scheduling novel approaches, the duty for making scheduling decisions are recline with one centralized scheduler, or by multiple distributed schedulers. In a novel Grid system, there are many applications submitted concurrently. The centralized approach shaves the advantage offense of implementation, but they have some shortcoming such as lack of scalability, fault tolerance. For example, Sabin et al [11] propose a centralized approach which uses backfill for scheduling jobs in multiple heterogeneous locations. In a similar manner, Arora et al [12] present a decentralized, dynamic scheduling and load balancing approach for the Grid systems. This approach uses a smart search strategy for searching partner nodes to which tasks can immigrate. It also extends over this decision making process with the accomplishment of ready jobs, thereby saving valuable processor cycles. In IHLBA[13], Yun et al send out the idea of load balancing to decrease completion time of jobs and to raise system performance. The algorithm selects suitable resource for jobs by taking into account the average computing power (ACP) of each resource. The ACP is obtained by using available CPU utilization and current CPU utilization. IHLBA then calculates the average load of each resource (ALC) and compares it with a balance threshold. If ALC is higher than balance threshold, the cluster is measured to be overloaded. The under loaded clusters with high ACP will be selected and jobs will be assigned to it.

Most of the novel grid scheduling is presented on the basis of the decentralized scheduling model where submitted jobs in a particular node are allocated to other nodes when the jobs cannot be adapted in that node [14]. Grids change consequently in size and the makespan is very dynamic. Makespan for scheduling algorithms is an important performance characteristic. Hence, theoretical worst-case analysis is a pertinent method as it presents these guarantees with performance bounds. In [15]Schwiegelshohn et al. proved that the performance of Garey and Graham's list scheduling algorithm is significantly worse in Grids than in multiprocessors.

From another point of view, Grid scheduling approaches can be separated into two types: online mode scheduling and batch mode scheduling. In the Batch mode job scheduling approaches, Jobs stands in a line and gathered together from different places in into a set when they reach in the batch mode. The scheduling starts after a fixed period time [16]. First come first served scheduling algorithm (FCFS) and Round robin scheduling algorithm (RR) are two of the most famous of Batch mode job scheduling approaches. In FCFS algorithm, jobs are executed under the order of job arriving time. The next job will be executed one by one. The RR algorithm principally focuses on the decency problem. This algorithm clarifies a ring as its queue and a fixed time quantum. Each job can be executed just within this quantum one by one [10][17].

The ORC scheduling comprises the best fit followed by round robin scheduling which deliver to the jobs between the processors which are ready for use. The remaining un-allotted jobs stand in the line for next execution. Therefore the order of jobs execution makes the performance of the processor better and distribute load efficiently across the system. This will decrease the waiting time of jobs in the queue and prevent the starvation [18]. In the On-line mode job scheduling algorithm, after jobs arriving, they will be scheduled. Because the Grid is a heterogeneous environment and it varies quickly, the on-line mode job scheduling algorithms are more suitable for the Grid environment [19]. Most fit task scheduling algorithm (MFTF) and Ant colony optimization (ACO) are two of the most famous of the On-line mode job scheduling approaches. MFTF principally try to discover the fitness between resources and tasks. It assigns resources to tasks under a fitness value [20]. Ant colony optimization (ACO) algorithm is on the basis of Ant algorithm and changed it to match the Grid system. It requires some information; for example number of CPUs, MIPS for each processor, etc. for scheduling the jobs [21].

III. Research Methodology

Considering the properties of P2P[22] and grid and their functional similarity; using the benefits of P2P in Grid can make better the performance of Grid. To find a solution for the single point failure in centralized models, P2P technologies have been used in grid environment in novel approaches. Because we use P2P in our approach, we chose JXTA [23] as an infrastructure of our approach. JXTA provides a great appliance for node grouping. Because JXTA uses two-layer architecture, the main search process performs only in super peers so the overhead of the network is decreased. We use a hybrid P2P structure[24] where the nodes are divided into some clusters; in each cluster there is a coordinator, which holds the information of all of the cluster nodes. Figure 1 shows the structure of our model.

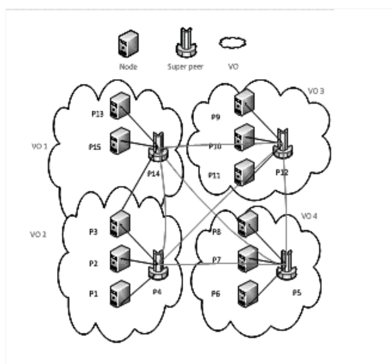

Figure 1 : The topology of our approach

IV. Theory

When the super cluster node receives a request from a user, it will compute the Average Processing Speed (APS) and Average Job Length (AJL) of all of its nodes. Then the scheduler who allocated in the super cluster assigns a premiership to the request. We named it Job Premiership (JP). The super cluster first computes deserve of its underlying nodes with the aid of fuzzy theory, and then it sends the job Premiership (JP) to the all of the super clusters. All of the super clusters computes deserve of its underlying nodes with the aid of fuzzy theory. The output value of the fuzzy system is between 0 and 1. The most suitable cluster will be selected according to the mentioned value. After finding the most suitable cluster, the underlying node with highest Processing Speed (PS) will be selected and jobs will be assigned to it.

Our approach is on the basis of fuzzy logic which uses some current status of the system as the input parameters of fuzzy system. These are some values which represent Average Processing Speed, Average Job Length and Job Premiership. Teases non-fuzzy values are the inputs of fuzzy inference system which is used for fuzzy reasoning step. The output value of fuzzy inference system has a non-fuzzy value which shows if the cluster has the adequate node for the request or not. Figure 2 shows the used fuzzy inference system.

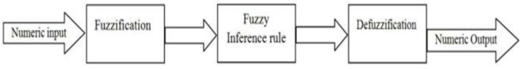

Figure 2: The used fuzzy inference system in our approach

There are two typical types of fuzzy inference systems; Mamdani and Sugeno[25]. Mamdani inference system is used in our approach because of its simplicity. When a request for scheduling is issued, it will be send to the super peer of the groups and then it should be directed to the cluster that is most suitable for the request. For selecting the suitable clusters, cluster which has the following two characteristics is selected; first, a suitable APS; Second, it should has low AJL. These two parameters are two of the input parameters for fuzzy system. Job Premiership is the third parameter of fuzzy system. In fact we use fuzzy theory with three parameters for selecting the most suitable cluster. Processing speed of the nodes is a parameter which should be considered in P2P grid approaches because typically the jobs launched in hybrid P2P grid overlays have low communications between nodes and high execution of time. Figure 3till 5 represent separately the fuzzy sets for APS, AJL and JP parameters which are made by using Matlab fuzzy logic toolbox.

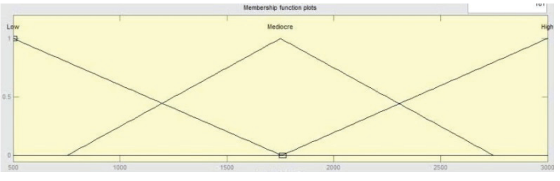

Figure 3: Fuzzy sets for APS

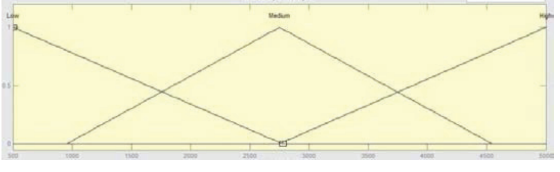

Figure 4: Fuzzy sets for AJL

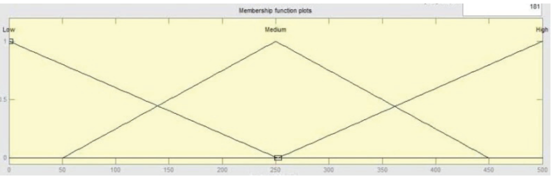

Figure 5: Fuzzy sets for JP

The fuzzy rules in Mamdani inference system can be produced according to the past experiences. In our approach we have used the following rules as shown in Table 1.

Table 1
Some of the rules which are used in our approach

Average Processing Speed (APS)	**Average Job Length (AJL)**	**Job PREMIERSHIP (JP)**	**Result**
Low	Low	Low	Mediocre
Low	Low	Medium	Mediocre
Low	Medium	Low	Adequate
Low	Medium	High	Mediocre
Low	High	Medium	Adequate
Low	High	High	Adequate
Medium	Low	Low	Mediocre
Medium	Low	Medium	Adequate
Medium	Medium	Low	Mediocre
Medium	Medium	High	Adequate
Medium	High	Medium	Adequate
Medium	High	High	Adequate
High	Low	Low	Inappropriate
High	Low	Medium	Mediocre
High	Medium	Low	Inappropriate
High	Medium	High	Mediocre
High	High	Medium	Inappropriate
High	High	High	Mediocre

Some rules will be fired based on the input parameters. The fired rules should be integrated in such a way that a decision made According to the aggregation of the fired rules. In Aggregation of fired rules step the fuzzy sets that represent the outputs of each fired rule are integrated into a single fuzzy set. This single fuzzy set is the input for the defuzzfication step. The output of defuzzification step is a non-fuzzy number which determines the most adequate cluster for the received request. Figure 6 shows the aggregation of fired rules and the defuzzification phase which is performed by Matlab fuzzy logic toolbox. A request is directed to a cluster which has the highest value in the output phase of fuzzy step. The selected super cluster obtains its underlying PS nodes; then it will be directed to the node which has the highest PS value.

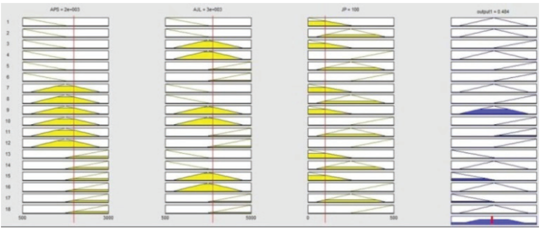

Figure 6: The process of aggregation of fired rules and deffuzzification step

The next pseudo-code algorithm shows the steps of our decentralized grid job scheduling approach:

```
Algorithm 1

Input: The job which requested by User
Output: Some nodes which are suitable for running the job.
Begin
Receive the job.
//The received job is sent to the closest super cluster.
The super cluster computes deserve of its underlying nodes with the aid of fuzzy theory.
//The deserve of the cluster will be obtained by fuzzy theory.
// the input of fuzzy system are APS, AJL and JP.
The super cluster sends the job to the all of the super clusters.
All of the super clusters computes deserve of its underlying nodes with the aid of fuzzy theory.
The most suitable cluster will be selected. // the super cluster with the higher value.
The selected super cluster computes deserve of its underlying nodes based on their Processing Speed PS value.
The node with the highest PS value is selected as the most suitable node.
The super cluster assigns the job to the selected node.
End
```

V. Results And Discussions

In this section we represent the results of the performed simulations. In these experiments the proposed approach is compared with other approaches in some evaluation criteria with Gridsim simulator [26] and Matlab[27].

For these experiments, the protocols of JXTA are not carried out in Gridsim. But for keeping some similarities with JXTA protocols, the message size of the Gridsim is considered as a proximate length in bytes of the JXTA headers and message length for considering these protocols. When a big Grid infrastructure is carried out in GridSim simulator, the low level protocols such as JXTA protocols are not essential to implement.

In the first experiment, our model has been evaluated in term of Makespan (response time) with HLBA and IHLBA [13]. We use some parameters as shown in table 2for comparing our approach in term of the makespan with HLBA and IHLBA, and we will compare it with other scheduling algorithms in the next experiment. As it is represent in table 3 and figure 7 our approach has the better performance due to considering both of processing speed value, current load of each node and the request premiership as the current status of the system simultaneously. It should be taken into account that APS is different of Average Computing Power (ACP). As it mentioned in [13], for computing ACP, CPU MIPS of resource and the current CPU utilization of the resource are considered; but for obtaining APS we only consider the processor computing value of each nodes. Similarly, Average Job Length (AJL) is different of average load of each cluster (ALC) [13]. For computing ALC, CPU utilization of the resource, the memory utilization of the resource and the utilization of network should be considered; but for obtaining AJL we only consider the current job lengths which are running on the nodes.

Table 2
Simulation parameters of the first experiment

Number of nodes of a cluster	Number of clusters	Number of Task	Computing power of resource node (MIPS)	Baud rate (bps)	Size of task (MI)
10	10	1000	500-5000	1500	200000-400000

Table 3
Makespan of our approach compared with iHLBA and HLBA

HLBA	IHLBA	New method	Plan
3500	3200	2000	Plan span (number of tasks 1000)

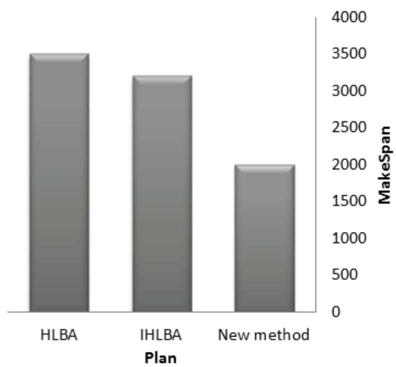

Figure 7: Makespan of our approach compared with iHLBA and HLBA

In the second experiment, our model has been evaluated in terms of makespan with some other algorithms. We compare our approach with iHLBA[13], ACO algorithm [28], MFTF algorithm [29] and random selection method. Table 4 shows the simulation parameters of this experiment.

Table 4
Simulation parameters of the second experiment

Size of task (MI)	Baud rate (bps)	Computing power of resource node (MIPS)	Number of tasks	Number of clusters	Number of nodes of a cluster
300000-500000	1500	500-5000	2000	10	10

As we mentioned earlier, our approach considers the essential current status parameters of the system simultaneously; so the makespan is very efficient rather than other approaches. It is true that iHlBA considers ALC and ACP for scheduling; but these parameters are not considered simultaneously. So we can achieve a better performance than other approaches. In iHLBA a cluster with highest ACP means that it is the most adequate cluster for the resource. The ACP of each cluster is calculated based on the newest status of resources. So iHLBA has the better performance rather than ACO, MFTF and random selection method in term of makespan. In ACO algorithm, if a resource finishes the assigned job, the chance of that resource for being chosen will grow in later job submissions; but it may raise the load of the better resource and increase the makespan. In MFTF algorithm, a job assigns to the resource with highest fitness value. MFTF also assigns job to the fastest resource and it neglects the load of the resource. So, the makespan has higher value than our approach. Random method assigns job to a resource randomly. It neglects the status of system and resources. So, the makespan of the random algorithm is the worst. Table 5 and Figure 8 represents the result of the second experiment.

Table 5
Makespan of our approach compared with other approaches

Plan	New method	IHLBA	MFTF	ACO	Random
MakeSpan	3250	4700	5800	6000	6300

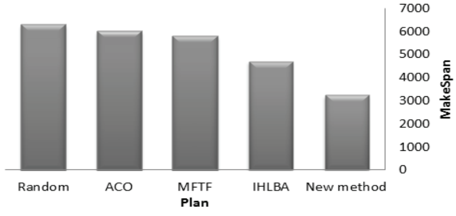

Figure 8: Makespan of our approach compared with other approaches

In the third experiment we calcite the performance of our approach in term of standard deviation of the load of clusters. Standard deviation of the load of clusters determines the gap of load between clusters. If an algorithm has a small standard deviation value, it means that algorithm has a balanced status. iHLBA defines a balance threshold which can make the balance in the loading of clusters better. This balanced threshold is not fixed; it is compatible based on the current status of resources. The ACO algorithm considers the load of system. In ACO algorithm, user defined variables have a major influence for standard deviation value; so it cannot have a god value in this criteria. MFTF and Random algorithms neglect the load of resources; therefore they have not a good value for this criterion. Our approach considers the current job length value and it is the load value for the nodes. Furthermore in this experiment, we set the same values for the weight of CPU utilization of the resource, the memory utilization of the resource and the utilization of network as they are defines in iHLBA. According to the simulation results, considering AJL and APS simultaneously, have a major influence on CPU utilization of the resources parameter and memory utilization of the resources parameter. These two parameters are the major parameters for calculating standard deviation of the load of clusters [13]. So, that is exactly why the presented approach can improve standard deviation of the load of clusters. Table 6 and Figure 9 represent the results.

Table 6
Standard deviation of the load of clusters withother approaches

Plan	New method	IHLBA	MFTF	ACO	Random
Standard deviation of the load of clusters with each method	23	25	25.5	26.5	27

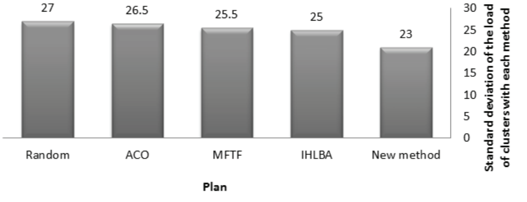

Figure 9: Standard deviation of the load of clusters with each method

VI. Conclusion

In recent years a lot of attention has been paid to the grid job scheduling. Grid Performance is based on an effective job scheduling procedure; hence, there is an essential for an effective approach to decrease response time. In this paper, a distributed novel approach for job scheduling in the Grid environment has been presented. Fuzzy theory is one of the intelligent approaches which present uncertainty. In our new approach, fuzzy is used for selecting the most adequate resources. Our approach with the aid of fuzzy theory allocates the most suitable resource to the job which considers the intermediate load of nodes of each clusters, the average of computing power and job premiership simultaneously as the input parameters of fuzzy system and selects the most suitable node base of the output value of fuzzy system. The fuzzy system uses a knowledge base of an expert(if-then).it takes three parameters simultaneously. And uncertainties (according to the dynamic Grid environments), which is based on mathematical principles.'s Exact acts. And continuity properties, distribution and variability .

We evaluated the performance of our approach with some grid job scheduling approaches; the results of the experiments confirmed the efficiency of the new approach in term of scalability and makespan (response time). The results show that the new approach improves the makespan because of considering the current system status.

REFERENCES

[1] C. Foster, I., Kesselman, "The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann," *San Francisco*, 1998.

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

[3] et al Ernemann, Carsten, "On advantages of grid computing for parallel job scheduling.," *Cluster Computing and the Grid*, 2002.

[4] S. E. D. R. A. Mosleh, Mohammad, Shahdad Shariatmadari, "A novel approach for grid resource management based on fuzzy logic and semantic Technology.," *International Journal of Innovative Computing*, 2013.

[5] M. M. Javanmardi, Saeed, Shahdad Shariatmadari, "A Novel Decentralized Fuzzy Based Approach for Grid Resource Discovery.," *International Journal of Innovative Computing*, 2013.

[6] K. H. Song, Shanshan, Yu-Kwong Kwok, "Security-driven heuristics and a fast genetic algorithm for trusted grid job scheduling," *Parallel and Distributed Processing Symposium*, 2005.

[7] A. A. Allahverdiyev, "Application of Fuzzy-Genetic Algorithm for Solving an Open Transportation," *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, vol. 3, no. 7, 2011,pp. 119–123.

[8] B. Y. 8. Klir, George J., "Fuzzy sets and fuzzy logic," *New Jersey: Prentice Hall*, 1995.

[9] et al Wang, Na, "Hierarchical Structure Resource Scheduling Model Based on Grid," 2012.

[10] R. Y. V. Hamscher, U. Schwiegelshohn, A. Streit, "Evaluation of Job-Scheduling Strategies for Grid Computing," *Proc. of GRID 2000 GRID 2000,First IEEE/ACM International Workshop Performance analysis of load balancing algorithms, World Academy of Science*,2000, pp. 191–202.

[11] P. S. G. Sabin, R. Kettimuthu, A. Rajan, "Scheduling of Parallel Jobs in a Heterogeneous Multi-Site Environment," *International Workshop on Job Scheduling Strategies for Parallel Processing, Lecture Notes In Computer Science*, vol. 2862, 2003.

[12] R. B. M, Arora, S.K. Das, "A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments," *International Conference on Parallel Processing Workshops (ICPPW'02)*, 2002,pp. 499 – 505.

[13] Y. Lee, S. Leu, and R. Chang, "Improving job scheduling algorithms in a grid environment," *Future Generation Computer Systems*, vol. 27, no. 8, 2011, pp. 991–998.

[14] K. Li, "Job Scheduling for Grid Computing on Metacomputers," *International Parallel and Distributed Processing Symposium (IPDPS 2005)*, 2005.

[15] R. Y. U. Schwiegelshohn, A. Tchernykh, "On-line scheduling in Grids," *IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1–10.

[16] G. V. L. He, XiaoShan, XianHe Sun, "QoS guided min-min heuristic for grid task scheduling.," *Journal of Computer Science and Technology*, 2003, pp. 442–451.

[17] K. H. Fujimoto, Noriyuki, "A comparison among grid scheduling algorithms for independent coarse-grained tasks," *Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium on. IEEE*, 2004.

[18] S. R. K.Somasundaram, "Node Allocation in Grid Computing Using Optimal Resource Constraint (ORC) Scheduling," *IJCSNS International Journal of Computer Science and Network Security,VOL.8 No.6*, 2008.

[19] C.-F. L. Ruay-Shiung, Chih-Yuan Lin, "Scheduling jobs in grids adaptively," *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on. IEEE*, 2009.

[20] M. O. Ghanbari, Shamsollah, "A Priority based Job Scheduling Algorithm in Cloud Computing," *Procedia Engineering 50*, 2012, pp. 778–785.

[21] P.-S. L. Chang, Ruay-Shiung, Jih-Sheng Chang, "An ant algorithm for balanced job scheduling in grids," *Future Generation Computer Systems*, 2009, pp. 20–27.

[22] H. G.-M. Kamvar, Sepandar D., Mario T. Schlosser, "The eigentrust algorithm for reputation management in p2p networks.," in *Proceedings of the 12th international conference on World Wide Web*, 2003.

[23] L. Gong, "JXTA: A network programming environment.," *Internet Computing, IEEE 5.3*, 2001, pp. 88–95.

[24] et al Loo, Boon Thau, "The case for a hybrid P2P search infrastructure," *Peer-to-Peer Systems III. Springer Berlin Heidelberg*, 2005, pp. 141–150.

[25] M. Sugeno, "An introductory survey of fuzzy control.," *Information sciences*, 1985, pp. 59–83.

[26] et al. Sulistio, Anthony, "A toolkit for modelling and simulating data Grids: an extension to GridSim.," *Concurrency and Computation*, 2008, pp. 1591–1609.

[27] S. N. Sivanandam, S. N., Sai Sumathi, "Introduction to fuzzy logic using MATLAB," *Springer*, 2007.

[28] J. S. Xu, Xiangdan Hou, "Ant algorithm-based task scheduling in grid computing," *Canadian Conference on Electrical and Computer Engineering 2*, 2003, pp. 1107–1110.

[29] Z.-Y. H. Sheng-De Wang, I-Tar Hsu, "Dynamic scheduling methods for computational grid environments," *International Conference on Parallel and Distributed Systems 1*, 2005, pp. 22–28.