A static jobs scheduling for independent jobs in Grid Environment by using Fuzzy C-Mean and Genetic algorithms

SIRILUCK LORPUNMANEE[1], MOHD NOOR MD SAP[2], ABDUL HANAN ABDULLAH[3], SURAT SRINOY[4]
[1,4]Faculty of Science and Technology, Suan Dusit Rajabhat University
295 Rajasrima Rd., Dusit, Bangkok, Thailand,
Tel: (662)-2445244, Fax: (662)-6687136
Email: {[1]siriluck_lor, [4]surat_sri}@dusit.ac.th http://www.dusit.ac.th

[1,2,3]Faculty of Computer Science and Information Systems, University Technology of Malaysia, 81310 Skudai, Johor, Malaysia,
Tel: (607) - 5532070, Fax: (607) 5565044
Email: { [2]mohdnoor, [3]hanan}@fsksm.utm.my http://www.utm.my

***Abstract*—The concept of Grid computing is becoming a more important for the high performance computing world. Such flexible resource request could offer the opportunity to optimize several parameters, such as coordinated resource sharing among dynamic collections of individuals, institutions, and resources. Specifically, we investigate the static job scheduling algorithm for independent jobs. In this paper we propose and evaluate experimentally a static scheduling for independent jobs that rely on determining job characteristics at runtime and jobs allocate to resources. We present a static job scheduling algorithm by using Fuzzy C-Mean and Genetic algorithms. Our model presents the strategies of allocating jobs to different nodes, which we have developed the model by using Fuzzy C-Mean algorithm for prediction the characteristics of jobs that run in Grid environment and Genetic algorithm for jobs allocated to large scale sharing of resources. The performance of our model in a static job scheduling have researchers will be discussed. Our model has shown that the scheduling system will allocate jobs efficiently and effectively.**

Key-Words: - Grid computing, Fuzzy C-Mean algorithm, Genetic algorithm, a static jobs scheduling, Heterogeneous computing system.

I. INTRODUCTION

Grid Computing is the principle that occurs for a long period of time by focusing on virtual organizations [1] to share large-scale resources, innovating applications and in some cases getting high-performance orientation. Under this principle, Grid has problem in flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources. In Grid [2] concept is a new generation of technologies combine physical resources and applications that provide vastly more effective solutions to complex problems (e.g., scientific, engineering and business). These new technologies must be built on secure discovery, jobs allocates to resources, integration resources and services from the others. In [3] is a formal definition of Grid concepts. They define conceptual models is abstract machines that support applications and services. Fig. 1 taken from [3], are formally defined (e.g., Organization, Virtual Organization, Virtual Machine, Programming System, etc.). Currently, Global Grid Forum [4] formulated and provided standards documents of virtual organization.

Grid Computing goes beyond distributing and sharing resources to more applications, which are adapting to use Grid resources. Although, using distributed resources are useful but this is possible only if the Grid resources are scheduled as well. The optimal scheduler will result in high performance in Grid computing but the poor scheduler will be making contrast result. Now, the Grid scheduling is big topic in Grid environment for new algorithm model**.**

The Grid scheduling is responsible for resource discovery, resources selection, and job assignment over a decentralized heterogeneous system, which resources belong to multiple administrative domains. Normally, the resources are requested by a Grid application, which use to computing, data and network resources etc. However, Grid scheduling of applications is absolutely more complex than scheduling an applications of a single computer. Because resources information of single computer scheduling is easy to get information, such as CPU frequency, number of CPU a machine, memory size, memory configuration and network bandwidth etc. But Grid environment is dynamic resources sharing and distributing. Then an application is hard to get resources information, such as CPU load, available memory, available network capacity etc. And Grid environment also hard to classify jobs characteristic, that run in Grid. There are basically two approaches to solve this problems, the first is based on jobs characteristic and second is based on a distributed resources discovery and allocation system. It should optimize the allocation of a job allowing the execution on the optimization of resources. The scheduling in Grid environment has to satisfy a number of constraints on different problems.

We have defined a set of them to study the feasibility and the usefulness of applying Fuzzy C-Mean algorithm and genetic algorithm to this field. The model is designed for multiple objectives in the jobs scheduling system in which often involve a lot of historical data and many complex objective. Our model consider in job submit time, run time, idle time and jobs end time that each job running on the Grid. Fuzzy C-Mean algorithm and Genetic algorithms are applied to solve the jobs scheduling system with multiple objectives and the results have shown that the jobs scheduling system can improve the performance and can allocate jobs efficiently and effectively.

In this paper we discussed Fuzzy C-Mean clustering and Genetic algorithm in scheduling system. To this end, we are reviewing a related work in Section II. Next, Section III we brief our design and architecture. Section IV, concludes the paper by summarizing this work, and providing a preview of future research in this area, is given in Section V.

II. RELATED WORKED

Most of the job scheduling in Grid environment based on job execute time and job run time has been proposed. In [5], the module prediction engine is a part of scheduling and offer a history based approach for estimating the run time of job submission. Intelligence will be a key feature in the next generation of Grid environment. In [6] proposed two modules for predicting the completion time of jobs in a service Grid and applying genetic algorithm to job scheduling. The problems of scheduling system on the Grid environment have been in [7], [8], [9], [10]. All of them adopt the method of genetic algorithm for jobs scheduling by applying different jobs characterization to improve performance. We noticed that their methods focused on optimization or sub-optimization for scheduling system.

Efficacious and effective job scheduling in Grid requires to model can allocate the available resources on Grid nodes to compute jobs, determine the current workload and predict the job execution time. In [11], was job scheduling in parallel and cluster computing; their goals are to achieve best performance and load balancing across the entire system.

Facing varying situations, intelligent Grid environments need complicated scheduling strategies and algorithms to handle different kinds of jobs.

Heuristic algorithms are often used in Grid environment for scheduling system. The algorithms use historical data of workload and explicit constraints to scheduling jobs [12], [13].

In [14] proposed models for scheduling system by using genetic algorithm with multiple goals. It's considered problems in parallel machine scheduling.

Our approach, especially examine the implications of the fact that workload of jobs is expected to have an impact on the resources utilization and, even more interestingly for researcher on the performance quality. We use information about static workload data from the Standard Workload Archive [14] and it has been experimented in several publications [15], [16]. These workload traces consists of information about all job submissions on a machine for a certain period of time which usually ranges over several months and several thousands of jobs. Therefore, it is reasonable to start with the available workload traces information from the compute centers to evaluate the impact of jobs characterization in Grid environment.

III. DESIGN AND ARCHITECTURE

A. Clustering

Clustering algorithms refer to automatic unsupervised classification method in data set and it can be classification into different categories data set base on the type of input parameters, type of clusters they are interested.

Clustering algorithms for data sets can be found in different fields, such as statistics, computer science, bioinformatics and machine learning. The most famous clustering algorithm is the Fuzzy C-Mean algorithm.

This paper aims to introduce cluster analysis for classification of jobs characteristic, or objects, according to similarities among them, and organizing objects into groups. A cluster is a group of objects that are more similar to each other than to objects in other clusters. Similarity is often defined by means of distance based upon the length from a data vector to some prototypical object of the cluster.

The data are typically observations of some phenomenon. Each object consists of *m* measured variables, grouped into an *m*-dimensional column vector $x_i = \{x_{i1}, x_{i2,...,} x_{in}\}$. A set of *n* objects is denoted by $U = x_i = \{x_{i1}, x_{i2,...,} x_p\}$

$$X = [x_{ij}] = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{p,1} & x_{p,2} & \cdots & x_{p,n} \end{pmatrix}$$

To distinguish between labeled and unlabeled patterns we will introduce a two-valued (Boolean) indicator vector $b = [b_k], k = 1, 2, ..N$ with 0 - 1 entries in the following manner

$$b_k = \begin{cases} 1 & \text{if pattern } x_k \text{ is labeled} \\ 0 & \text{otherwise} \end{cases}$$

A-1. Fuzzy c-means (FCM) clustering

The Fuzzy C-Means is one of the existing clustering methods for building fuzzy partitions. This method will be used in this paper as the basic tool for building jobs characterization in Grid environment. Fuzzy C-Means (FCM) algorithm, also known as fuzzy ISODATA, was introduced by Bezdek [15] as extension to Dunn's [16] algorithm to generate fuzzy sets for every observed feature.

Fuzzy clustering methods allow for uncertainty in the cluster assignments. Rather that partitioning the data into a collection of distinct sets (where each data point is assigned to exactly one set), fuzzy clustering creates a fuzzy pseudo partition, which consists of a collection of fuzzy sets. Fuzzy sets differ from traditional sets in that membership in the set is allowed to be uncertain. A fuzzy set is formalized by the following definitions.

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of given data, where $x_i \in \Re^n$ is a set of feature data. The minimization objective function of FCM algorithm is frequently used in pattern recognition follow as;

$$J(U,V) = \sum_{j=1}^{C} \sum_{i=1}^{N} (\mu_{ij})^m \|x_i - v_j\|^2 \quad ; 1 \le m < \infty \quad (1)$$

Where m is any real number greater than 1, $V = \{v_1, v_2, ..., v_C\}$ are the cluster centers. $U = (\mu_{ij})_{NxC}$ is the degree of membership of vector x_i in the cluster j. The values of matrix U should satisfy the following conditions;

$$\mu_{ij} \in [0,1] \; ; \; \forall_i = 1,2,\ldots,N \; ; \; \forall_j = 1,2,\ldots,C \quad (2)$$

$$\sum_{j=1}^{C} \mu_{ij} = 1 \; ; \; \forall_i = 1,2,\ldots,N \quad (3)$$

The Euclidean distance $d_{ij} = \|x_i - v_j\|$ is any norm expressing the similarity between any measured data and the center and calculate the cluster centers $\boldsymbol{V}$ according to the equation:

$$v_j = \frac{\sum_{i=1}^{N} (\mu_{ij})^m x_i}{\sum_{i=1}^{N} (\mu_{ij})^m} \quad ; \forall_j = 1,2,\ldots,C \quad (4)$$

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update the fuzzy partition matrix U by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}} \quad (5)$$

This algorithm will stop if $\max_{ij} \left\| \mu_{ij}^{(k+1)} - \mu_{ij}^{(k)} \right\| < \phi$; $\phi \in [0,1]$ and k is the iteration step. All equation shown above, converge to Fuzzy C-Mean algorithm by following step:

Algorithm Fuzzy *C*-Means
Step 1: Generate an initial *U* and *V*
Step 2: At k-step: calculate the centers vectors $C^{(k)}=[v_j]$ with $U^{(k)}$:

$$v_j = \frac{\sum_{i=1}^{N} (\mu_{ij})^m x_i}{\sum_{i=1}^{N} (\mu_{ij})^m}$$

Step 3: Update $U^{(k)}$, $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

Step 4: If $\| U^{(k+1)} - U^{(k)} \| < \phi$ then STOP; otherwise return to step 2.

Figure 1: Fuzzy *c*-means [17]

B. Genetic Algorithms

Genetic algorithms (GA) are a class of stochastic search algorithms which based on biological evolution [18], [19]. A basic GA can be represented as in Fig 7.2 taken from [20].

Genetic algorithms combine the exploitation of past results with the exploration of new areas of the search space by using survival of the fittest techniques combined with a structured of randomized information exchange, a GA can mimic some of the innovative intelligence of human search. A generation is a collection of artificial creatures (strings). In every new generation a set of strings is created using information from the previous times. Occasionally a new part is tried for good measure. GAs are randomized, but they are not simple random walks. They efficiently exploit historical information to speculate on new search points with expected improvement.

The approach used in this work generates a set of initial scheduling, evaluates the scheduling to obtain a measure of fitness, selects the most appropriate and combines then together using operators (crossover and mutation) to formulate a new set of solutions.

The basic type of GAs, known as the simple GA (SGA), uses a population of binary strings, single point crossover, and proportional selection [18], [19]. Many other modifications to the SGA have been proposed; some of these are adopted in our work. The following subsections explain the steps in our proposed approach;

GA( )
{
Generate initial population of Jobs individuals
Evaluate individuals according to fitness function;
While *stopping condition is satisfied.*
{
Count from 1 to amount generation;
Select two parents from initial population
(Population $\longrightarrow$ *Parent*$_1$ *and Parent*$_2$*);*

```
  Crossover (Parent1 and Parent2) ⟶ Child;
  Mutation (Parent p, Parent q, Child);
  Fitness (Child c, Best Chromosome bc);
  Improvement (Child c);
  Replace (Chromosome chrom, Child c);
  Scheduling(best chromosome);
 }
} return set of the best chromosome in population for
  job scheduling.
```

C. Design and Algorithm

This section will briefly describe how workload data will separate to three classifications and the strategic of them will allocate to optimal Grid nodes. Let us consider a set of jobs of workload, a set of heterogeneous computing system. Jobs are subject to constraints on the Grid environment then we selected a set of properties of jobs and descriptions relevant to real world situations, an example dataset is presented as in Table 1.

Table 1:data set of jobs properties representation

Instance	Properties type	Decision field
1	Job Number	Yes
2	Submit Time	No
3	Run Time	Yes
4	Number of allocated processors	No
5	User ID	Yes
6	Group ID	No

Let $J = [j_{xy}] = \begin{bmatrix} j_{11} & j_{12} & \cdots & j_{1y} \\ j_{21} & j_{22} & \cdots & j_{2y} \\ \vdots & \vdots & \vdots & \vdots \\ j_{x1} & j_{x2} & \cdots & j_{xy} \end{bmatrix}$ be a set of given jobs, where y is the yth of User ID and x is the xth of Job Number. As a Grid environment consists of the vector value of n nodes and a node consists of m resources then following as:

$$\vec{G} = \begin{bmatrix} \vec{N}_1 \\ \vec{N}_2 \\ \vec{N}_3 \\ \vdots \\ \vec{N}_n \end{bmatrix} \text{ and } \vec{N} = [R_1 \quad R_2 \quad R_3 \quad \ldots \quad R_m]$$

Then $\vec{G} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ R_{n1} & R_{n2} & \cdots & R_{nm} \end{bmatrix}$

Fig. 2 shows our model by using Fuzzy C-Mean clustering technique for prediction jobs characteristic to three classifications, first Heavy jobs workload, second Medium jobs workload, finally Low jobs workload. And we designed GA algorithm, which was the strategies of allocating jobs to different nodes.

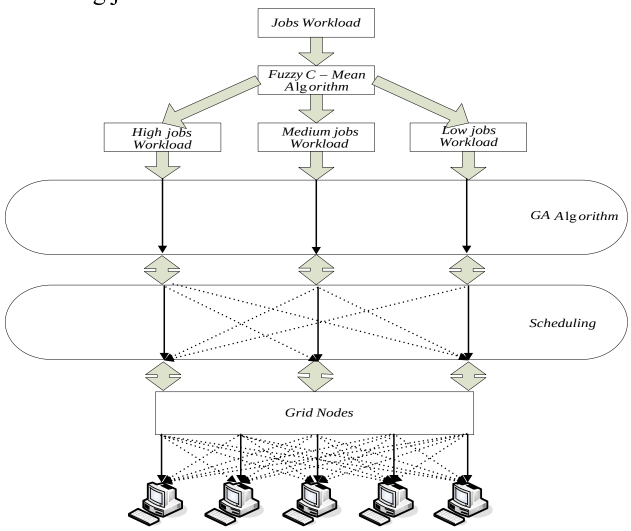

Figure 2: Our Model

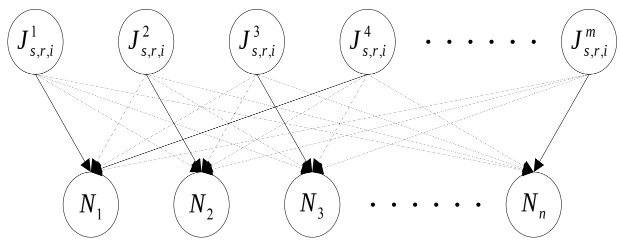

Figure 3: The strategic of allocation jobs to Grid nodes

Fig. 2, 3 shows how the job will compute to optimal Grid nodes, which we separate jobs to three classifications, after that we allocate each jobs in different classification to optimal Grid nodes.

D. Our scheduling algorithms

Fuzzy C-Mean and SJF( ) {

Input: $J = [j_{xy}] = \begin{bmatrix} j_{11} & j_{12} & \cdots & j_{1y} \\ j_{21} & j_{22} & \cdots & j_{2y} \\ \vdots & \vdots & \vdots & \vdots \\ j_{x1} & j_{x2} & \cdots & j_{xy} \end{bmatrix}$,

where as each of jobs order in job number.

Find classification of workload by using Fuzzy C-Mean, such that

1. Heavy workload: Group of jobs is large job run-time. Such as $t_{H1}, t_{H2}, \ldots, t_{Ha}$; $1 \le a \le xy$
2. Medium workload: Group of jobs is medium job run-time. Such as $t_{M1}, t_{M2}, \ldots, t_{Mb}$; $1 \le b \le xy$
3. Light workload: Group of jobs is short job run-time. Such as $t_{L1}, t_{L2}, \ldots, t_{Lc}$; $1 \le c \le xy$

while (jobs ≤xy) {
 if (Heavy workload) {
  Mapping jobs to processor by using GA:

$$t_{H1} \to N_{best}$$
$$t_{H2} \to N_{best}$$
$$t_{H3} \to N_{best}$$
$$\vdots$$
$$t_{Hx} \to N_{best}$$

 }
 if (Medium workload) {
  Mapping jobs to processor by using GA:

$$t_{M1} \to N_{best}$$
$$t_{M2} \to N_{best}$$
$$t_{M3} \to N_{best}$$
$$\vdots$$
$$t_{Mx} \to N_{best}$$

 }
 if (Light workload) {
  Mapping jobs to processor by using GA:

$$t_{L1} \to N_{best}$$
$$t_{L2} \to N_{best}$$
$$t_{L3} \to N_{best}$$
$$\vdots$$
$$t_{Lx} \to N_{best}$$

 }
 }
}

Output: $WaitingTime_{j_1} = WaitingTime_{j2} = WaitingTime_{j3} = \ldots = WaitingTime_{jxy}$

Min(Makespan$_n$), where as n is each of node in Grid environment.

$$Min\left(\frac{\sum_{j=1}^{xy} WaitingTime_j}{xy}\right)$$

IV. EXPERIMENTAL SETUP AND RESULTS

In this experiment, we used jobs workload data from the Standard Workload Archive [18]. This data consists of 1,000 jobs, each of which has 18 properties field, however we focused on some properties that previous mention. In our experiments we assumed that each of jobs is allowed to run in each node by using space-sharing mechanism. Our simulation, we simulated 50 different performance nodes in Grid environment.

Our experiments showed classification of jobs workload to three groups, first is heavy workload, second is medium workload and third is light workload.

Figure 4 shows the jobs characterization by using Fuzzy C-Mean algorithm separated workload data to three groups, (a) is heavy workload, (b) is medium workload and (c) is light workload.

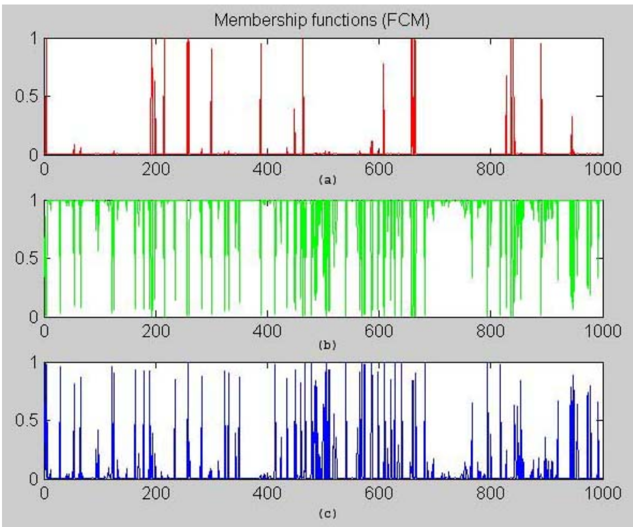

Figure 4: Jobs characterization

In this experiment, the jobs in workload data is predicted to three classification, heavy workload is Run-Time from 6,959 to 19,761 and total jobs are 22 jobs, Medium workload from 1,790 to 6,540 and total jobs are 69 jobs and Light workload from 1 to 1,759 and total jobs are 909, see Table 2.

Table 2: jobs classifications in Grid environment.

No.	Classification	Total
1	Heavy Workload	22
2	Medium Workload	69
3	Light Workload	909

In Figure 5, 6 shows the distribution of make-span time and average waiting time of all jobs in Grid environment. Our simulation define crossover rate (P_c) is 0.9, mutation rate (P_m) is 0.9 and 500 generation with each type of workload.

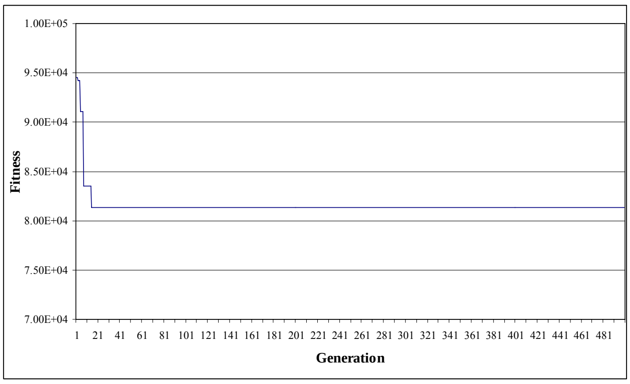

Figure 5: Make-span time of all jobs in Grid environment

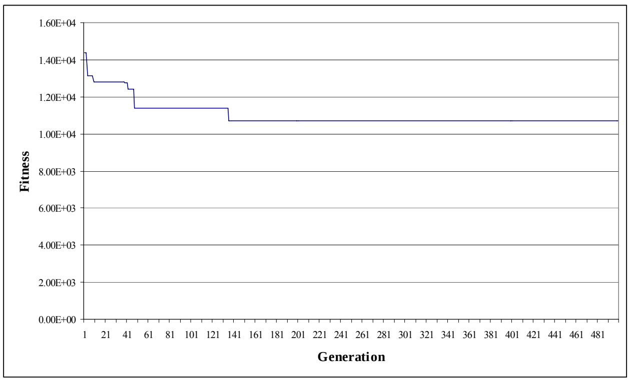

Figure 6: Average waiting time of all jobs in Grid environment

V. CONCLUSIONS

We have studied the job scheduling problem for Grid environment as a combinatorial prediction and optimization. Several observations are listed;

1. We have proposed the intelligence scheduling in Grid environment and used it for our algorithm by using Fuzzy C-Mean clustering technique for predicting three classifications and GA algorithm for allocate them to different Grid nodes.
2. We used jobs workload from the Standard Workload Archive [18] on space-sharing mechanism. We show that the proposed model captures the jobs characterization of real workload in three different classifications. This model can be used to our algorithm for simulating and evaluating scheduling policies for simulating Grid environment.
3. Many scheduling algorithms for Grid environment depend on static information provided by the Standard Workload Archive [18] and the different performance nodes in Grid, simulating by us. We observe that this information is often unreliable. However, it is a useful way.
4. The experiment results on the make-span time and average waiting time have shown that the scheduling system using our algorithm can allocate the optimal results.
5. The results provided here suggest that the researcher look forward to new method for such problems should consider combine them with their method.

VI. FUTURE WORK

The following will be our future works;

1. Our simulation environment will include critical parameters, such as submit time, Grid network cost, job migrations overhead, faults tolerance.
2. We plan to investigate the swarm intelligence mechanism, such as ant colony algorithm for Grid scheduling.
3. We will include a more complex characterization of the constraints for Grid scheduling and will improve the complexity problems in Grid environment.

VII. ACKNOWLEDGMENT

We would like to thank Suan Dusit Rajabhat University for supported us.

REFERENCES

[1] Foster I, Kesselman C, Tuecke S. The anatomy of the Grid: Enabling scalable virtual organizations. International Journal of Supercomputer Applications 2001.
[2] I. Foster and C. Kesselman, Eds., The Grid 2: Blueprint for a New Computing Infrastructure. San Francisco, CA: Morgan Kaufmann, 2004.
[3] M. Parashar, J. Browne, Conceptual and Implementation Models for the Grid.
[4] Global Grid Forum [Online]. Available: http://www.gridforum.org.
[5] ArshadAli, Ashiq Anjum, Julian Bunn, R. Cavanaugh, Frank van Lingen, R. McClatchey, Muhammad Atif Mehmood, H. Newman, C. Steenberg, M. Thomas, I. Willers. PREDICTING THE RESOURCE REQUIREMENTS OF A JOB SUBMISSION.
[6] Y. Gao, H. Rong, Joshua Zhexue Huang. Adaptive grid job scheduling with genetic algorithms.
[7] Downey, A, Predicting Queue Times on Space-Sharing Parallel Computers, in International Parallel Processing Symposium, 1997.
[8] Gibbons, R., A Historical Profiler for Use by Parallel Schedulers, Master's thesis, University of Toronto, 1997.
[9] Vraalsen, F., R.A. Aydt, C.L. Mendes, and D.A. Reed, Performance Contracts: Predicting and Monitoring Grid Application Behavior, GRID 2001: Proceedings of the Second International Workshop on Grid Computing, 2001.
[10] MEHRA, P., SCHULBACH, C. H., AND YAN, J. C. A Comparison of Two Model-Based Performance-Prediction Techniques for Message-Passing Parallel Programs. In Proceedings of the ACM Conference on Measurement & Modeling of Computer Systems - SIGMETRICS'94 (Nashville, May 1994), pp. 181–190.
[11] SAAVEDRA-BARRERA, R. H., SMITH, A. J., AND MIYA, E. Performance prediction by benchmark and machine characterization. IEEE Transactions on Computers 38, 12 (December 1989), 1659–1679.
[12] KAPADIA, N., FORTES, J., AND BRODLEY, C. Predictive Application-Performance Modeling in a Computational Grid Environment. In Proceedings of the Eight IEEE Symposium on High-Performance Distributed Computing (Redondo Beach, California, August 1999), pp. 47–54.
[13] PETITET, A., BLACKFORD, S., J.DONGARRA, ELLIS, B., FAGG, G., ROCHE, K., AND VADHIYAR, S. Numerical Libraries and The Grid: The GrADS Experiments with ScaLAPACK. Tech. Rep. UT-CS-01-460, University of Tennessee, April 2001.
[14] Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/, Juni 2004.
[15] J. Bezkek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, USA, 1981.
[16] J.C. Dunn, "A Fuzzy Relative of the ISODATA process and its Use in Detecting Compact", Well Separated Clusters, Journal of Cybernetics, Vol. 3, No.3, 1974, pp. 32-57.

[17] A Tutorial on Clustering Algorithms: "Fuzzy C-Means Clustering" http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/index.html.

[18] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riordan. Modeling of Workload in MPPs. In D. Feitelson and L. Rudolph, editors, IPPS'97 Workshop: Job Scheduling Strategies for Parallel Processing, pages 94–116. Springer–Verlag, Lecture Notes in Computer Science LNCS 1291, 1997.

[19] C. Ernemann, B. Song, and R. Yahyapour. Scaling of Workload Traces. In D. G. Feitelson, L. Rudolph, and U. Schwiegelshohn, editors, Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003 Seattle, WA, USA, June 24, 2003, volume 2862 of Lecture Notes in Computer Science (LNCS), pages 166–183. Springer-Verlag Heidelberg, October 2003.

[20] J. Krallmann, U. Schwiegelshohn, and R. Yahyapour. On the Design and Evaluation of Job Scheduling Systems. In D. G. Feitelson and L. Rudolph, editors, IPPS/SPDP'99 Workshop: Job Scheduling Strategies for Parallel Processing. Springer, Berlin, Lecture Notes in Computer Science, LNCS 1659, 1999.