Improving Grid Computing Scheduling using Heuristic Algorithms

Mohamed Eisa
Computer Science Department, Port Said University
42526 Port Said, Egypt.
eisa@informatik.uni-freiburg.de,
mmmeisa@yahoo.com.

ABSTRACT
In this paper, the proposed model schedule user tasks using multiple scheduling algorithms based on grid computing middleware. The presented proposed model based on two resolution methods. This allowed us to increase utilization of global scheduler and decrease makespane at local scheduler. This has been applied on grid computing simulator and experimental results indicate that heuristic algorithms decrease waiting time at global scheduler in grid architecture.

Keywords
Grid computing; heuristic algorithms; scheduling.

1. INTRODUCTION
Interest in grid computing [1, 2] has grown considerably because of their applications in modern technology and industries, such as Resource management, QoS, networking, storage, security, adaptation, policy, etc. The choice of algorithms used to schedule jobs in such environments depends on the target application. Scheduling strategically allocates the CPU to a process based on specified criteria. There are many different methods of selecting which process will be given control of the CPU. Each of these methods follows a different scheduling algorithm and has advantages and disadvantages .A series of studies on grid computing has been initiated by many authors; in particular, In [3] M. Maheswar et al propose an opportunistic load balancing (OLB) or myopic algorithm, where the resource with the currently shortest waiting queue or the smallest waiting queue time is selected for the incoming task. This approach is its simplicity, but it is often far from optimal.

In recent years, there have been several studies [4, 5, 6] on hybrid scheduling systems that combined meta heuristic search algorithms with scheduling algorithms. These algorithms depend on multiple criteria; many of these criteria have been suggested for comparing CPU scheduling algorithms such as CPU utilization, throughput, waiting time, etc.

J. Schopf [7] Introduces scheduling process in the grid into three stages: resource discovering and filtering, resource selecting and scheduling according to certain objectives, and job submission. B. A. Shirazi et al [8] assume a system model in which many users submit requests for job execution from any one of a large number of sites. At each site an external scheduler (ES), responsible for determining where to send jobs submitted to that site and a local scheduler (LS), responsible for determining the order in which jobs are executed at that particular site. The choice of algorithms for each component defines a particular scheduling system. This model doesn't consider which algorithm is appropriate at each site, where the numbers of jobs at external scheduler are greater than the other at local scheduler. In the last decade this filed of research concerned with improving the performance of grid computing by using hybrid scheduling algorithms using heuristic algorithm that increase the utilization of resources , reduce searching space and makespane time.

In this paper, the proposed model composed of global scheduler with its global queue and local scheduler also with its queue. In the last decade, this filed of research concerned with improving the performance of grid computing by using hybrid scheduling algorithms using heuristic algorithms that increase the utilization of resources, reduce searching space and makespane time.

The remainder of this paper is organized as follows: section 2 describes the preliminaries and notations. Model construction is then presented in section 3, followed by a description of the experimental design and the results are presented and discussed in section 4. The paper concludes in section 5.

2. PERLIMINARIES AND NOTATIONS

2.1 Grid computing

Grid computing is a collection of computers that intended to accomplish great tasks and allow multiple users from different platforms to access network resources and execute their tasks. M. Baker et al [9], define grid computing as model that aggregate multiple resources from different geographically distributed systems to enhance availability, capability, performance, cost and user's quality of services. Grid scheduling is the process of selection and allocation of resources between administrative domains.

Scheduling process consisting of several elements are shown in Figure 1. This Figure refers to the mechanism of mapping resources to different tasks at a single site or multiple sites based on a user scheduling algorithm. Scheduler must make a selection of resources in an environment where you have no control over local resources, the resources are distributed and the system information is often limited. The schedule of the task include: maximize system performance by maximizing the use of resources, reduce the execution time and compliance with financial constraints.

In addition, the local resource manager providing services to start a task in a state of the given resource verification works and retrieve the results when the task is finished.

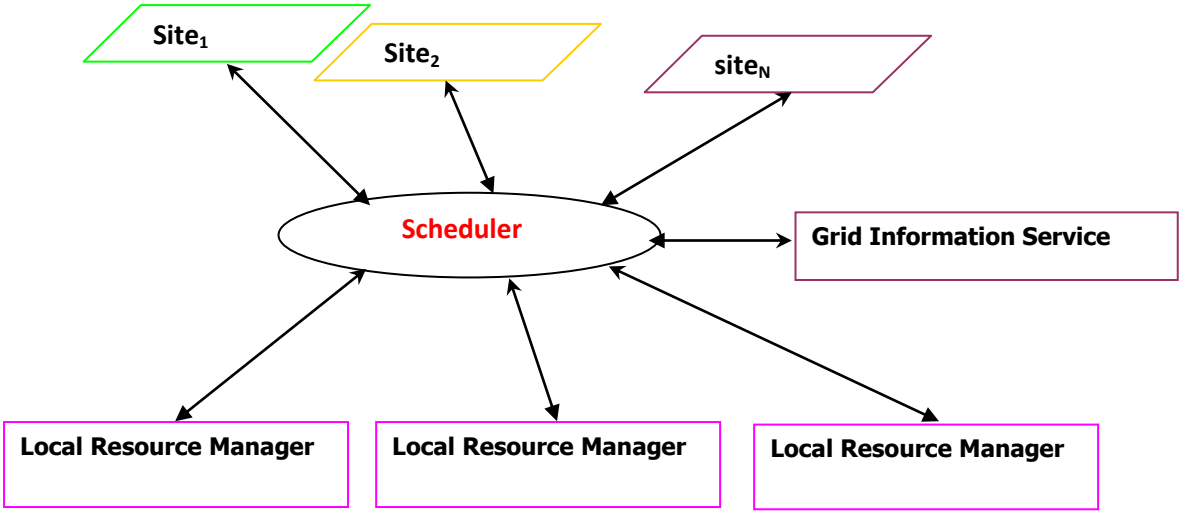

Fig. 1. Grid Scheduling Process.

3. MODEL CONSTRUCTION

Grid scheduling aims to maximize system performance, increase resource utilization and minimize execution time. Therefore, we introduce the model shown in Figure 2. Tasks are assigned to local scheduler according to their priority in the task's queue. We assume a model in which many users submit their tasks for the implementation from various sites. This model is illustrated in Figure 2, which sent requests (R1, R2, …, Rn) from different sites accepted by Global Scheduler (GS) queue. Each request is sent via the communication network checks its local correction and then forwarded to the local queue. Consider a model in which many users send requests to perform work of a whole of a large number of sites.

At each site, two components are located: (j) a global scheduler, responsible for determining where to send tasks that sent to this place, (ii) a Local Scheduler (LS) which determine the order in which the work is performed at each site. Heuristic algorithms [10, 11] used to improve the scheduling process and optimize resource mapping. Two heuristic algorithms are presented in this work to improve the utilization of resources and the execution time of the task.

In the first, we introduce Genetic Algorithm (GA) as a global search technique to select the optimal solution from the search space. General form of a GA depicted in Figure 3, where Pi relates to the population of resources and Evaluate is a function determining the quality of individuals in the population (usually inversely dependent on their objective function values).

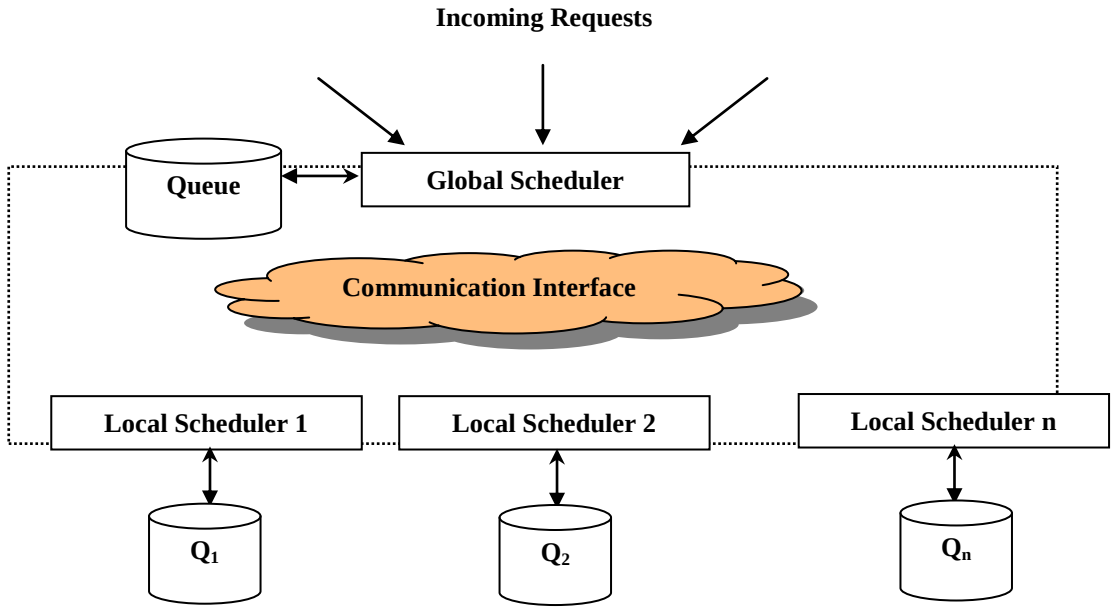

Fig. 2. Grid resources and scheduling.

Second, Game-based scheduler: For solving the Grid users' problem using Minimum Completion Time (MCT) algorithm introduced as a tool for scheduling problems in grid computing. General form of a MCT depicted in Figure 4, where the main scheduler mechanism at the global level is defined as a GA-based scheduler where a task is assigned to the machine yielding the earliest completion time (defined as the sum of ready time for the machine and time of computing all tasks assigned there). The process is repeated until remain tasks to be assigned.

1. Initialize(i, Pi);
2. Evaluate(Pi);
3. repeat
4. Select Resources (Pi);
5. Recombinate(Pi);
6. Evaluate(Pi);

Fig. 3. Genetic algorithm pseudo-code.

1. For all tasks in meta task Mt .
2. For all resources Ri .
3. Do until all tasks mapped.
4. Find the task with minimum competition time.
5. Find the resources with the earliest competition time.
6. i := i + 1;
7. until Stop_Criterion;
8. Assign the recourse Ri to the task Mt .
9. Until stop.

Fig. 4. MCT pseudo-code.

4. EXPERIMENTAL RESULTS

Traditional scheduling algorithms can not be optimally managed grid resources. Thus, we used heuristic algorithms to improve the output of scheduling algorithms by increasing utilization of resources and reducing search space. In this section, we carried out our experiments based on HyperSim-G Grid simulation package to evaluate the performance of different combinations of task and task scheduling algorithms [12]. HyperSim-G has been developed for the Grid environments. Assume that all processors have the same performance and each user generates tasks according to some distribution. Table 1 specifies the simulation parameters used for our study. We use 32 machine have the same processing capacity in this section. Under this consideration, we present a series of simulation results of the model described above. Total numbers of tasks are 512 tasks. Users are mapped evenly across sites and submit tasks according to a Poisson distribution with an inter-arrival time of 5 seconds.

Table 1: Experimental parameters

Parameters	Values
Total number of tasks	512
Machines in the system	32
Managing tasks at host	FCFS
Grid scheduling	Genetic Algorithm
Population size	100
Crossover probability	0.8
Selection choice	Random
Mutation probability	0.4

Results indicate that our approach increase utilization of global scheduler and decrease makespane at local scheduler. This has been applied on grid computing simulator and the experimental results indicate that heuristic algorithms decrease waiting time at global scheduler in grid architecture. As shown in Figure 5 and figure 6 waiting time for local and global scheduler under different system utilization factors, respectively. Observation indicated that GA increase server utilization by increase the total busy time and decrease the total ideal time. The reason is that the GA decreases the waiting time at global scheduler.

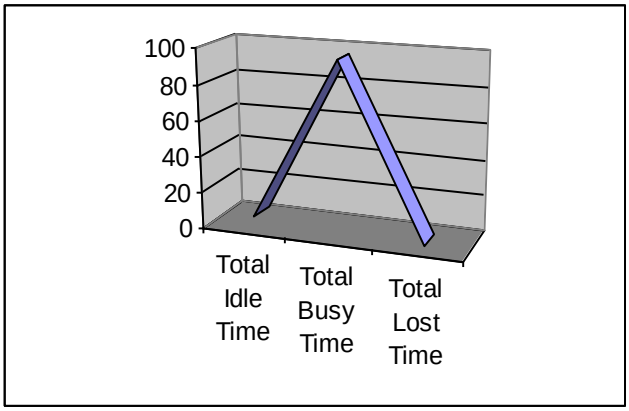

Fig. 5. Average response time for different combinations of scheduling and host utilization using genetic algorithm.

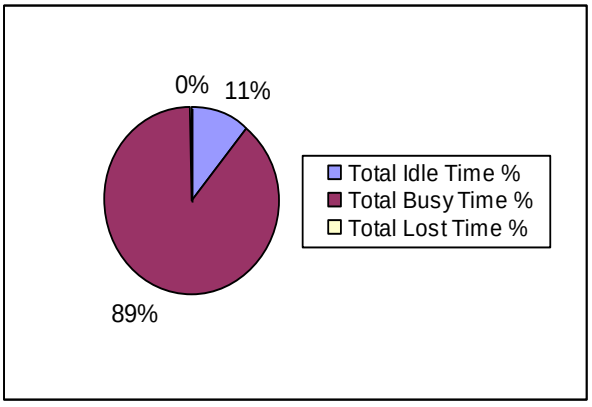

Fig. 6. Average response time for different combinations of scheduling and host utilization using Game-based scheduler.

On the other hand, we observe that GA improves the utilization of resources and the execution time of the task rather than Game-based scheduler.

5. CONCLUSIONS

In this paper, we are interested in two distinct functionalities: global scheduling and local scheduling. We consider a model composed of global scheduler with its global queue and local scheduler also with its queue. The model focuses in increase the resource utilization at global queue and decrease makespane time at local queue. The objective of this work is to investigate the mechanisms of scheduling problems in grid computing and to find whether scheduling algorithms used at global and local queue are similar.

6. REFERENCES

[1] Ian Bird, Bob Jones, Kerk F. Kee, "The organization and management of grid infrastructures", IEEE Computer 42(1), 2009, pp. 36_46.

[2] J. Li, "Grid resource discovery based on semantically linked virtual organizations", Future Generation Computer Systems 26 (1), 2010, pp. 361_373.

[3] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen and R. F. Freund, "Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems", in J. of Parallel and Distributed Computing, Vol. 59, No. 2, pp.107--131, November 1999.

[4] C. Castillo, G. N. Rouskas, and K. Harfoush, "Efficient resource management using advance reservations for heterogeneous grids," in IPDPS'08: IEEE International Parallel and Distributed Processing Symposium. ACM, 2008, pp. 1–12.

[5] A. Kertesz, et al., "GMBS: a middleware service for making grids interoperable", Future Generation Computer Systems 26 (1), 2010, pp. 542_553.

[6] A. Abraham, R. Buyya and B. Nath. Natures Heuristics for Scheduling Jobs on Computational Grids, The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000) India, 2000.

[7] J. Schopf, "Ten Actions When Super Scheduling, document of Scheduling Working Group, Global Grid Forum", http://www.ggf.org/documents/GFD.4.pdf, July 2001.

[8] B. A. Shirazi, A. R. Husson, and K. M. Kavi, " Scheduling and Load Balancing in Parallel and Distributed Systems". IEEE Computer Society Press, 1995.

[9] M. Baker, R. Buyya and D. Laforenza, "Grids and Grid Technologies for Wide-area Distributed Computing", in J. of Software-Practice & Experience, Vol. 32, No. 15, pp. 1437-1466, December 2002.

[10] I. Foster, C. Kesselman, " The Grid: Blueprint for a New Computing Infrastructure, second ed"., Morgan-Kaufman, Los Altos, CA, 2004.

[11] L. Zhang,"Scheduling algorithm for real-time applications in grid environment", in: Proceedings on IEEE International Conference on Systems, Man and Cybernetics, vol. 5, USA, 2002.

[12] HyperSim-G Grid simulation package at: http://weboptserv.lsi.upc.edu/.