

A Survey of Job Scheduling Algorithms With Hierarchical Structure to Load Balancing In Grid Computing Environments

Maryam Masoudi Khorsand
Departement of
Computer Engineering
Islamic Azad University
Sciences and Research Branch
Ardabil, Iran

Shahram Jamali
Department of
Computer Engineering
University of
Mohaghegh Ardabili
Ardabil, Iran

Morteza Analoui
Department of
Computer Engineering
University of
Science and Technology
Tehran, Iran

Abstract: Due to the advances in human civilization, problems in science and engineering are becoming more complicated than ever before. To solve these complicated problems, grid computing becomes a popular tool. a grid environment collects, integrates, and uses heterogeneous or homogeneous resources scattered around the globe by a high-speed network. Scheduling problems are at the heart of any Grid-like computational system. a good scheduling algorithm can assign jobs to resources efficiently and can balance the system load. in this paper we survey three algorithms for grid scheduling and compare benefit and disadvantages of their based on makespan.

Keywords: Grid Computing; Load Balancing; Hierarchical; Scheduling; Makespan.

1. INTRODUCTION

Computational grids as a new approach to solve large-scale problems in science, engineering and business have been most focused on large scale resource sharing, cooperation of several organizations and their use in new applications. Thus Grid computing systems involve a collection of programs and resources that are distributed among the machines of the grid. [1]. The goal of grid computing is to manage the system such that set the jobs to be done in shortest time. one of the aspects that improve the efficiency and decrease the time to performance, is the jobs Scheduling. the concept of scheduling jobs this is that to be specified jobs when and in which machine be performed. in grid environment, scheduling and effective management of resources is vital and important. appropriate scheduling causes the one of the best service to users and applications, and from the other side from all the grid resources, and occur optimal and maximum utilization. the other hand fair load divided into between computers in network is from goals and benefits in grid computing. dividing the load on the grid means that the grid management system for any assigned application, select the best source for run an program assigned to the that computer to runing. therefore, by jobs effective scheduling using Appropriate distribution of the applications among processor resources, the total load will be divided among all computers.[2]

In this paper be reviewed scheduling algorithms with hierarchical structure that overall goal in all of them is increase the efficiency and reduce the makespan for grid computing.

2. BACKGROUND

2.1 Grid Computing

The grid is a network of computers and supercomputers is capable of much higher speed data processing made it possible. In other words, the execution time of jobs that would

take hours to cut a few seconds. Grid computing is a new technology that the goal is to share resources for the job to come. In computing grid, job scheduling is a very important problem. A good scheduling algorithm can assign jobs to resources efficiently and can balance the system load. Grid computing has tried everything so far in the field of high performance computing, including cluster computing, peer-to-peer computing, which is done brought together.

2.2 Resource Discovery In Computational Grid

In grid environment, resource is a reusable entity that serves to fulfill a job or request is. Resource can be a car or any other service network (a combination of machines, networks, and software) is. The grid does not have any limitations in terms of geographic area coverage and the resources available in a grid of geographically distributed, heterogeneous, and may also be placed in different locations have the resources belong to different organizations. shared resources in a Grid environment are May be a type of desktop systems, clusters, large data sets, storage devices, etc.

When a remote user to a remote resource to run or access a data the resource is required. The mechanism that the grid provides the structure must be able to appropriate resource for the user to discover. Thus one of the most basic jobs in a Grid is support of resource discovery mechanism. The purpose of discovery of resources, identify a list of available resources to assigning jobs. When a user requests a resource in the Grid environment to run a specific application, a method of discover the resource must be able to discover resources to user requests with low cost and time for the user so the user can quickly to the resource uses. Grid resource discovery algorithms in different ways to use it. Based on resource discovery are three models: centralized, distributed and

hierarchical. Went on to is a summary of the hierarchical model, we evaluate the algorithms, this paper benefited of this model.

2.3 Model of Hierarchical Resource Discovery

In this model, there is a central resource management system and a number of local resource management system and a central resource management system with local resource management systems to discover of resources interact does. One way Interact from the central management of to local schedulers, as shown in Figure 1.

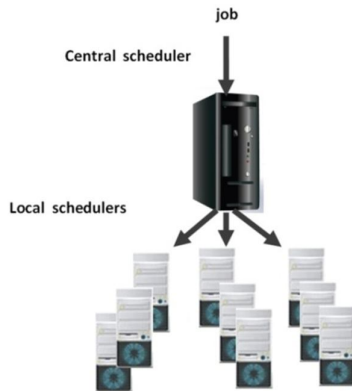


Figure1. Hierarchical scheduling model

As the centralized model, this model has problem of extensibility and fault-tolerance in that is low. But compared with the centralized model, an advantage of the hierarchical model is the central resource management and local resource management systems may have different policies.

2.4 Strategies of Load Balancing Algorithms

For the proper distribution of users requests, load balancing is very important in the grid environment. The load balancer receives users request and sends them to suitable resources. according to some specific distribution logic. Generally speaking, load balancing can be classified as static, dynamic, and adaptive. In static load balancing, the decision regarding the allocating resources to requests is predefined. In dynamic balancing, the request allocation decisions are decided at run time based on the current state of the system. In adaptive balancing, the allocation parameters and polices can vary depending on the run time information about the system, such as the current state of the system, previous decisions, etc [3,4].

3. REVIEWS OF HIERARCHICAL SCHEDULING ALGORITHMS

In the literature, many scheduling algorithms have been proposed. Most of them can be applied to the grid environment with suitable modifications. In general, they can be separated into two types: batch mode and on-line mode. In

this section, we will introduce some scheduling algorithms in on-line mode.

3.1 On-line Mode Heuristic Scheduling

Algorithm

Jobs are scheduled when they arrive. Since the Grid environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for the Grid environment.

3.1.1 Most Fit Job Scheduling Algorithm (MFTF)

MFTF algorithm tries to make jobs compatible with resources for job transmitter and assigns the resource to transmitter based on fitness function. [5]

Fitness function is calculated as follows:

$$\text{fitness}(i,j) = \frac{10000}{1+|W_i/S_j - E_i|} \quad (1)$$

W_i is load of i th job S_j is CPU speed in j th node. E_i is expected time of i th job. W_i/S_j is the performance time used in the given node. $W_i/S_j - E_i$ is the performance time of predicted jobs. E_i is identified by machines or transmitters:

$$E_i = A + n \times S \quad (2)$$

If A is mean responding time for 100 done jobs. N is nonnegative cardinal number and S is standard deviation of responding time for 100 done jobs. When performance time is measured near E_i , it means that this node is more suitable for job to be assigned. Since some processors may have much load and others remain free, this algorithm improves the efficiency of distributed system through load coordination and process capability of the system to smooth the periods with load traffic in the nodes. This is done by transmitting some loads from the nodes with heavy load to other processors in order to process.

Although many problems related to scheduling are solved with this algorithm, it can occur in the real environments with wrong scheduling because this algorithm doesn't consider the efficiency of resources.

3.1.2 Dynamic Load Balancing Algorithm (DLBA)

In [6] the authors propose an scheduling algorithm that performs an intra-cluster and inter-cluster load balancing. This algorithm in load index and other common parameters of each node in dynamic scheduling, the jobs. Intra-cluster load balancing is performed depending on the cluster manager cluster manager decides whether to begin the local balancing based on the current workload of the cluster of its child nodes. Inter-cluster load balancing is performed when some cluster manager unsuccessful to balance their workload. The local balancing unsuccessful may be due to a saturation of the cluster. In this position the cluster jobs excessive loaded,

according to selection strategies, has will be transferred to another cluster which is underloaded.

to justify the cluster is overloaded or not, they were introduced threshold called a threshold balance as Ψ . If the load of cluster more than Ψ , load balancing will be executed. This algorithm is feasible and improves the performance of the system.

However, the value of balanced threshold is fixed and set by its cluster, but the balanced threshold may not be suitable for the dynamic environment in the Grid system.

3.1.3 The Hierarchical Load Balancing

Algorithm (HLBA)

This algorithm, adaptive load balancing between clusters. Since assume jobs are computing-intensive jobs, and consider the computing power of each resource as the standard for the selecting resources.

Therefore, the system can assign the job to the appropriate resource according to the updated information that arrived from information server. The Information Server discovers resource nodes registered with the system, and records the information of the resource, such as CPU speed, idle CPU percentage, memory utilization, average load of each cluster, etc.

The Portal provides an interface for users to submit jobs. The job scheduler accepts the job from the portal and uses the HLBA to choose the proper cluster and compare its load with the system.

Then, it selects the resource with the integrates computing power in the cluster to execute the submitted job. After the job is finished, the result and the new status of the resource will be sent back to the Information Service for another scheduling[7].The system framework is shown in Figure 2.

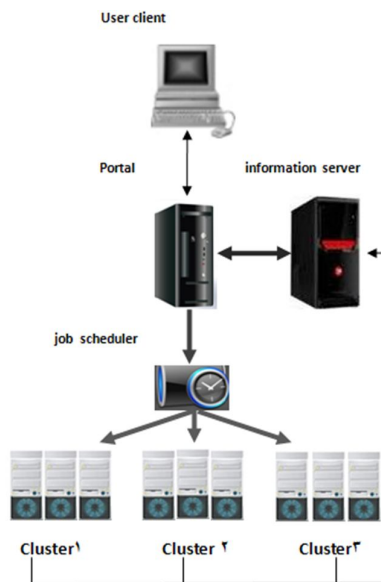


Figure2. system structure for HLBA

When the scheduler receives a job submitted by a user, it will transfer a request to the Information Service in order to obtain the necessary information such as the idle CPU percentage of each resource, average load of each cluster and average load of the system. Then the scheduler chooses a cluster which has the fastest average computing power (ACP). The average computing power of the cluster is defined as:

$$ACP = \frac{\sum_{k=1}^n CPU_SPEED_k \times (1 - CPU_k)}{n} \quad (3)$$

where CPU_Speed_k is the CPU MIPS of resource k in cluster i ; CPU_k is the current CPU utilization of the resource k in the cluster i , expressed as a percentage, and n is the number of resources in cluster i .

After the scheduler selects the cluster which has the fastest ACP, it will compare the average load of the chosen cluster with the average load of the system. The average load of the cluster is defined by the average load of each resource in cluster i . $Load_{k,i}$ represents the load of each resource k in the cluster i .

In HLBA, consider three load attributes, CPU utilization of the resource (CPU_k), the memory utilization of the resource (MU_k) and the utilization of network (NU_k). So the $Load_{k,i}$ may become:

$$Load_{k,i} = \sqrt{\alpha_1 CPU_k^2 + \alpha_2 NU_k^2 + \alpha_3 MU_k^2} \quad (4)$$

where $a1$ is the weight of the load attribute CPU_k , $a2$ is the weight of the load attribute NU_k , and $a3$ is the weight of load attribute MU_k . The average load of each cluster i is defined as

$$ALC_i = \frac{1}{n} \sum_{k=1}^n Load_{k,i} \quad (5)$$

The average load of the system is defined as:

$$AL = \frac{1}{m} \sum_{i=1}^m ALC_i \quad (6)$$

the average load of each cluster i , ALC_i , to be less than the balance threshold of the system. Hence, a threshold called balance threshold, denoted as Ψ and defined as below:

$$\Psi = AL + \sigma \quad (7)$$

where σ is the standard deviation of the load of the system and defined as below:

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - \bar{x})^2}, \quad \text{for all } i \quad (8)$$

where x is equals AL and x_i is equals ALC_i , and N is the number of clusters in the system.

When a job is to be assigned to a cluster with the highest ACP_i , the load of the selected cluster will be checked first to see if it is already overloaded. If the cluster's average load is more than Ψ , the cluster is marked as overloaded and the job will seek the cluster with the next highest ACP. If a cluster is

assigned a job, local update will perform. Namely recalculate ACP_i , $Load_{k,i}$, ALC_i , AL , and Ψ immediately. It will also affect the global update. But the average computing power and average load of other clusters are not affected. Therefore, the computation cost is less than a global update described below. The overall flowchart of the system is shown in Figure 3.

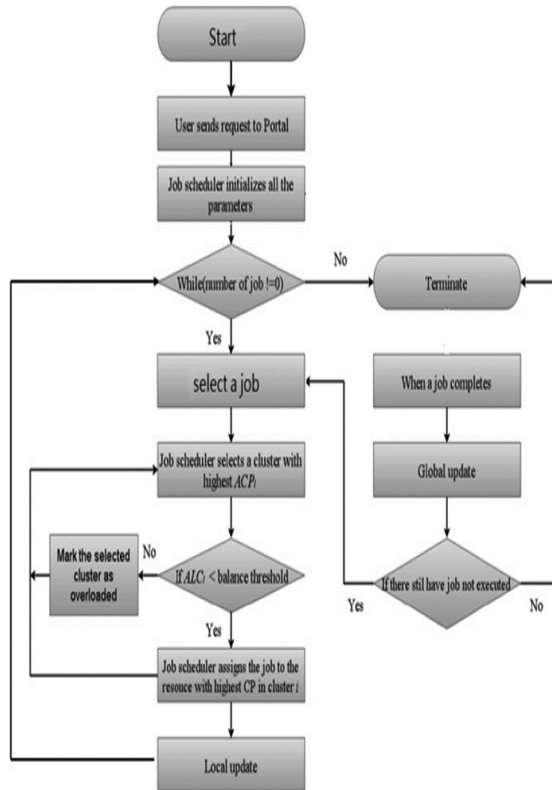


Figure3. Flowchart of HLBA

When a job is completed, the job had to release all resources and a global update will recalculate all Parameters again. Overloaded clusters can become available again after a global update.

3.1.4 Improvement of Hierarchy Load Balancing Algorithm (IHLBA)

This algorithm is an improved algorithm from HLBA. In this algorithm use the same parameters for scheduling but with different orders in HLBA. In HLBA, first choose the cluster with highest average computing power than compare its average load with Ψ . Considering Figure 3 there is a loop while selecting the cluster. This step may be the loss of time and impact on the makespan. when scheduler receives a job and obtains necessary information from the job, will sort clusters by their average loads. If the average load of cluster exceeds Ψ , means that the cluster is overloaded. In IHLBA sort the clusters which are underloaded and choose the cluster with the maximum ACP within those clusters. After choosing the proper cluster, then select the resource with the best computing power in this cluster and assign the job. Local

update and global update are also done in IHLBA [7]. If jobs have dependences and/or communications, their precedence relationship graph must first be drawn. Then a linearization is performed on this precedence graph. Jobs are then scheduled according to this linear order. If two jobs need to be executed at the same, they can be scheduled together as one job. The overall flowchart of the system is shown in Figure 4.

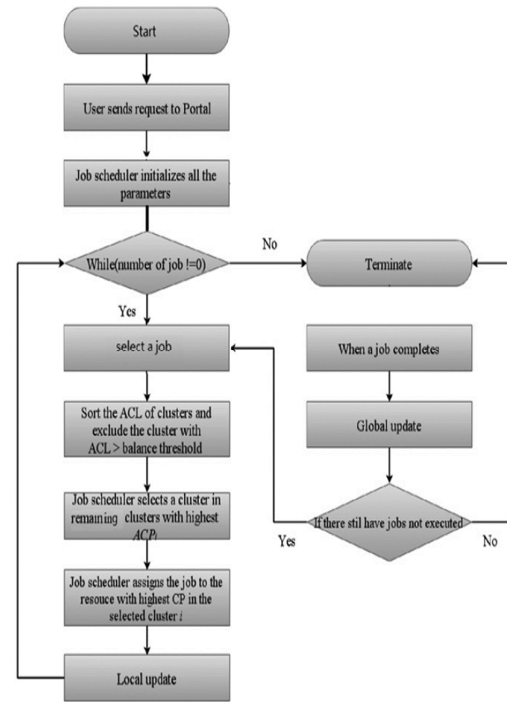


Figure4. Flowchart of IHLBA

assume that the cluster is static in this paper. Namely, the resources in a cluster are static. A dynamic cluster can also be handled by this algorithm since the dynamicity will be reflected in the cluster parameters. Another question is how the range of average computing power of clusters might affect the performance of this algorithm.

also, when a job arrives, the scheduler selects a cluster which is ALC Less than balance threshold and assigns the job to the resource with the maximum computing power in the cluster.

At the beginning of scheduling, the job scheduler will assign jobs to the resources in this cluster because it has the more average computing power than others. As time goes by, the load of the cluster becomes heavy and the ALC of the cluster exceeds the balance threshold. Therefore, in IHLBA algorithm may be removed when the cluster is overloaded, however is a cluster with high throughput. Thus, this clusters because has a little overload is not allowed to participate in competition clusters for resource allocation.

4. THE COMPARISON OF MENTIONED SCHEDULING ALGORITHM

In Table 1 Scheduling algorithms listed are compared for load balancing and makespan. The benefit and the disadvantages of for each is checked.

Table1. comparison of scheduling algorithms

parameters	advantages	disadvantages
MFTF Algorithm	It is compatible. It has high resistance in dynamic environments.	It doesn't consider the efficiency of resources.
DLBA Algorithm	This algorithm causing to load balancing.	Balance threshold may not appropriate for the characteristics of the dynamic grid systems.
HLBA Algorithm	This algorithm causing to load balancing and is appropriate for grid dynamic environments	In the time computing, it is not optimum.
IHLBA Algorithm	Cause Load balancing is and the ratio of the HLBA use optimum of time.	Due to few Overload may be to ignore the high processing resources

5. CONCLUSION AND FUTURE STUDIES

In this paper, four scheduling algorithm based on load coordination whit hierarchical structure in the grid systems were investigated. although there have been a lot of studies concerning job scheduling in grid systems, a new challenge still can be interesting and many research projects can be done. The present study was an attempt to focus on current scheduling algorithms and the mentioned algorithms were compared considering a variety of aspect and advantages and disadvantages of each one were explained. In the future, these algorithms can be tested on heterogeneous processors.

6. REFERENCES

- [1] Arora, S.K, Das, R and Biswas A. 2002. Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments, in Proc. Of International Conference on Parallel Processing Workshops (ICPPW'02), pp. 499 – 505
- [2] Braun, R., Siegel H., Beck N, Boloni .L and Maheswaran M. 2001. A Comparison of Eleven Static Heuristics for Mapping a class of Independent Jobs onto Heterogeneous Distributed Computing Systems, Journal of Parallel and Distributed Computing, pp. 810-837.
- [3] Casavant, T.L. and Kuhl J.G. 1988. A taxonomy of scheduling in general purpose distributed computing system, IEEE Transaction on Software Engineering 14 (2) 141–154.
- [4] Mehta, H, Kanungo P and Chandwani, M. 2008. Performance enhancement of scheduling algorithms in web server clusters using improved dynamic load balancing policies, in: 2nd National Conference, INDIACom-2008 Computing For Nation Development, New Delhi, Feb, pp. 651–656.
- [5] Wang S.D., Hsu I.T. and Huang. Z.Y. 2005. Dynamic scheduling methods for computational grid environments, International Conference on Parallel and Distributed Systems pp 22–28.
- [6] Suri P.K. and Manpreet S. 2010. An efficient decentralized load balancing algorithm for grid, in: IEEE 2nd International Advance Computing Conference, IACC, pp. 10–13.
- [7] Lee, Y.H., Leu, S and Chang, R.S. 2011. Improving job scheduling algorithms in a grid environment, Future Generation Computer Systems 27, pp.991-998.