Job Scheduling Algorithms in Grid Computing – Survey

G. Jaspher W. Kathrine
Assistant Professor,
School of Computer Science and Technology,
Karunya University,

Mansoor Ilaghi U
School of Computer Science and Technology
Karunya University,

Abstract

Grid computing is the form of distributed computing where the resources of various computers are shared to solve a particular problem. Grid can be used for variety of purposes. Job scheduling is used to schedule the user jobs to appropriate resources in grid environment. Here in this paper survey of various job scheduling algorithms are done. The job scheduling algorithms are compared and contrasted based on the makespan, flowtime, resource utilization and completion time.

1. Introduction

Grid computing [1] enhances computing facilities over internet. Grid computing has emerged from distributed computing where the resources of many computers in a network are used are used to solve a single problem at the same time. The difference between the conventional high performance computing such as cluster computing and grid is that grid tends to be loosely coupled, heterogeneous and geographically distributed. Though grid can be dedicated for a particular application, it can also be used for a variety of purpose. Search for Extraterrestrial Intelligence (SETI) is an example of Grid Computing Application.

Job scheduling is based on the necessity of a user who has a set of jobs to execute. Since the user's machine is not able to process the jobs either because of resource or hardware constraints, the user can use the grid system for running the job. The user submits the set of jobs to the job scheduler and the job scheduler splits the job depending on certain factors and gives it to the machines having available resources on the grid. The machines will complete the task and final result will be given to the user. In order to achieve this job scheduling strategies has to be followed.

Job scheduling and resource scheduling are the two main necessities in grid computing. In job scheduling, the job scheduler has to find the appropriate resource for the job that the user submits [2]. It has to find the best machine in grid to process the user job. Grid has two main schedulers such as local schedulers and grid schedulers. The local schedulers work in local computational environment and hence it is reliably, fast connection, works in uniform environment and also takes full control of the homogeneous resources [3]. Grid Schedulers also called as meta-schedulers are the top level schedulers. They are responsible for orchestrating resources that are managed by different local schedulers [4].

Scheduling can also be classified into static and dynamic scheduling. In static scheduling, before execution the jobs are assigned to the suitable machines and those machines will continue executing those jobs without interruption. In dynamic scheduling, the rescheduling of jobs is allowed. The jobs executing can be migrated based on the dynamic information about the workload of the resources [5].

In grid, there may be lots of resources to run a job. The main focus is to find the appropriate resource for the job that is to schedule the job. The methods for job scheduling are centralized, hierarchical and decentralized. In centralized scheduling, there will be a centralized scheduler and it is responsible for scheduling the jobs. It is very useful when all the resources have same objective. In hierarchical, there will be central scheduler. All jobs will be submitted to the central scheduler. The central scheduler redirects the jobs to the global scheduler. In decentralized, there is no central scheduler. Distributed schedulers coordinate with each other to schedule jobs [6].

The well-known techniques for job scheduling like Min – Min, Genetic Algorithm, Simulated Annealing Algorithm [7] [8] are in existence. In this paper, heuristic Min-Mean technique, Bio inspired techniques such as firefly and ant colony algorithm, grouping based technique and community aware scheduling algorithms are discussed in the following sections.

The paper has been organized as follows. In section 2, new enhanced heuristic min-mean scheduling is discussed. In section 3, firefly algorithm is explained. In section 4, ant colony algorithm is described. In section 5, adaptive grouping based job scheduling is explained. In section 6, the community aware scheduling algorithm is explained. Finally, in section 7, the conclusion and future work are presented.

2. Heuristic Min-Mean Job Scheduling

Online mode and batch mode are the two classifications of heuristic scheduling algorithms [9] [10]. In online mode, the jobs are scheduled to the resources as soon as it arrives. In batch mode, the jobs are independent, there is no order of execution and jobs are scheduled as a batch every time. Here in this paper, batch mode scheduling is followed. Achieving the minimum makespan is the goal.

To evaluate the mapping heuristic, the expected time to complete [ETC] model is employed. Before execution, the expected execution time of the tasks on the machine should be known, and this is contained in the ETC matrix. Consider for the task ti and the arbitrary machine m_j ETC[t_i,m_j]. This represents the expected time of the task i on the machine j. In this matrix, the row represents the expected execution time of a task on different machine and column represents the expected execution time of different tasks on the same machine[11][12][13].

Based on 3 characteristics, the benchmark of instances for distributed heterogeneous computing system is generated.

(i) Machine heterogeneity (low/high)
(ii) Task heterogeneity (low/high)
(iii) Consistency (Consistent/Inconsistent/Partially Consistent)

Combining these 3 bench marks, 12 combinations of ETC matrices are used to evaluate the heuristic min-mean scheduling algorithm.

There are 2 phases in this algorithm [14]. In the first phase, all jobs are assigned to the resources. In the second phase, mean completion time of all jobs is calculated and the jobs are allocated to the machines whose completion time is less than the mean completion time. Machine who has maximum completion time is selected as makespan.

3. Firefly Algorithm

The firefly algorithm is based on swarm intelligence behavior of firefly. It is a meta heuristic algorithm inspired by the social behavior of firefly. Firefly algorithm finds the global optimal solution. The main focus of firefly algorithm is to complete the task within a minimum makespan and flowtime as well to utilize the grid resource efficiently. Firefly optimization as mentioned in [15] [16] in can be described as

(i) The firefly attracts and is attracted by all other fireflies
(ii) The brighter one attracts the less brighter one
(iii) The brightness decreases with distance
(iv) The brightest firefly can move randomly
(v) The firefly particles can move randomly

There are 4 phases in firefly algorithm [17]

In the phase 1, the parameters are set (initial population, fitness and attractiveness), number of available resources and list of submitted jobs are identified.

In the phase 2, the brightness of each firefly is found at the source using fitness function and distance is calculated. The less bright fireflies are moved towards the brighter one.

In the phase 3, the new solution is evaluated and light intensity is updated

In the phase 4, the fireflies are ranked and current global best is identified. Finally, the iteration parameters are updated

All these are done until the termination condition is reached. The termination condition may be number of iteration or the fitness value or sometimes the saturation state.

4. Ant Colony Algorithm

The ant colony algorithm for job scheduling in grid aims at submitted jobs to resources based on the processing ability of jobs as well as the characteristics of the jobs.

Ant colony algorithm is the bio-inspired heuristic algorithm, which is derived from the social behavior of ants. Ants work together to find the shortest path between their nest and food source. When the ants move, each ant will deposit a chemical substance called pheromone. Using this pheromone, the shortest path is found. The same concept is used to assign jobs in grid computing. When a resource is assigning a job and completes, its pheromone value will be added each time. If a resource fails to finish a job, it will be punished by adding less pheromone value. The issue here is the stagnation, where there is a possibility of jobs being submitted to same resources having high pheromone value.

In this ant colony algorithm [18], the load balancing method is proposed to solve the issue of stagnation. The algorithm is as follows

(i) The user will send request to process a job
(ii) The grid resource broker will find a resource for the job
(iii) The resource broker will select the resource based on the largest value in the pheromone value matrix
(iv) The local pheromone update is done when a job is assigned to a resource.

(v) The global pheromone update is done when a resource completes a job

(vi) The execution result will be sent to the user

When the resource broker select a particular resource for a job j, jth column of the Pheromone Value matrix will be removed and jobs will be assigned to other resources. Thus the load balancing is achieved.

5. Grouping Based Job Scheduling in Grid Computing

The adaptive group based job scheduling focuses on group based scheduling, and explains the jobs are grouped based on coarse grained jobs.

The grouping is based on the resources' processing capability in Million Instructions per Second (MIPS), bandwidth (Mb/s), and memory size (In Mb).

The characteristics of resources are the basic for grouping strategy. In grid computing, 2 approaches can be used for job execution. In first approach, the user can directly search the resources for job execution using an information service. In the second approach, the user can obtain information about the current availability and capability of resources using the resource manager.

The Grouping Based Job Scheduling algorithm has 2 phases [19]. In the first phase, the scheduler receives information about the resource status from the Grid Information Service (GIS), and sorts the jobs in descending order. In the second phase, the system selects jobs in First Come First Served (FCFS) order and forms different job groups. The scheduler will select resource in FCFS order after sorting them in descending order of their MIPS. The jobs are put into the job groups until the sum of resource requirement of the jobs in that group is less than or equal to amount of resource available at selected site.

As soon as the job group is formed, the jobs are assigned to the corresponding resource. After execution the job groups, the result is sent to the corresponding user and the resources are available to the Grid System.

6. Community – Aware Scheduling Algorithm (CASA)

CASA is a decentralized dynamic heuristic metascheduling algorithm. In CASA, jobs can be rescheduled. In order to overcome the stagnation a probabilistic approach has been used to assign jobs so that the jobs are evenly distributed to all other resources.

CASA is a two phase algorithm [20]. The first phase is the job submission phase where each node receives the jobs that are submitted by local user. Consider a node A, it receives the job, it acts as a initiator node and requests all other nodes using the REQUEST message. The other nodes who are willing to take the job will reply through ACCEPT message. The node A will evaluate the other participating nodes using the historic data and selects the appropriate node and submits the job to it.

The second phase is the dynamic rescheduling phase, the node which received the job will look for the job which has large enough waiting time and has not been selected recently in the local job queue. That job will be rescheduled to the other nodes.

5 algorithms are discussed in CASA. They are:

(i) Job distribution
(ii) Job delegation request acceptance
(iii) Job assignment
(iv) Job rescheduling
(v) Job rescheduling request acceptance

Based on the above analysis of the algorithm, the algorithms are tabularized in Table 1 with advantages and disadvantages.

Table 1: Comparative Analysis of the algorithms

	PARAMETERS/ **PAPERS**	**ADVANTAGES**	**DISADVANTAGES**
1	**Heuristic Min-Mean Job Scheduling**	The main focus of minimizing the makespan is achieved	1)One resource can execute only one job at a time 2)Size and number of resources are static and should be know in prior
2	**Firefly Algorithm**	1)Position of the firefly is updated 2)Convergence and cost minimization	1)First come first served policy is followed 2)Initial population is fixed
3	**Community – Aware Scheduling Algorithm (CASA)**	1)Stagnation problem is solved through probabilistic approach 2) Has got strong adoptability	1)Message overhead
4	**Grouping Based Job Scheduling in Grid Computing**	1)Resource utilization is achieved 2)Improves processing of the fine grained jobs	1)Time complexity is high 2)Dynamic factors such as priority, network delay and QoS has to be taken into account
5	**Ant Colony Algorithm**	1)Adaptability 2)Good robustness in dynamic environment	1)Information on each resources is needed 2)The pheromone value has to be updated every time which may result in high processing time

7. Conclusion

Grid computing can solve complex tasks in shorter time and utilizes the hardware efficiently. To make the grid work efficiently, best job scheduling strategies have to be employed. Job scheduling is the foremost step in grid computing where the users' jobs are scheduled to different machines. The various strategies have been studied and classified. The advantages and disadvantages of the algorithms have been studied. The future work will be concerned with the development of the better scheduling algorithm which is heterogeneous and works in dynamic environment.

8. References

[1] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 15(3):200, 2001.

[2] R.Buyya and M.Murshed, "Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," Concurrency and Computation: Practice and Experience, vol. 14, pp. 1175–1220, 2002.

[3] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra and Prachet Bhuyan, "A Survey of Job Scheduling and Resource Management in Grid Computing", in World Academy of Science, Engineering and Technology 40, 2010.

[4] E. Huedo, R.S. Montero, and I.M. Llorente. The GridWay framework for adaptive scheduling and execution on grids. Scalable Computing: Practice and Experience, 6(3):1–8, 2005.

[5] M. Chtepen, "Dynamic scheduling in grids system," Sixth Firw PhD Symposium, Faculty of Engineering, Ghent University, pp. 110, 2005.

[6] Mr. Rakesh Kumar and Navjot Kaur , "Job Scheduling in Grid Computers,"

[7] Wang Meihong, and Zeng Wenhua, 2010. "A comparison of four popular heuristics for task scheduling problem in computational grid ", IEEE.

[8] Qinma Kang, Hong He, Hongrun Wang, and Changjun Jiang, 1996. "A Novel Discrete Particle Swarm Optimization Algorithm for Job Schedulingin Grids", in IEEE Fourth International Conference on Natural Computation, pp.401-405.

[9] Kun-Ming Yu, and Cheng-Kwan Chen., 2008. "An Adaptive Scheduling Algorithm for Scheduling Tasks in Computational Grid", in IEEE Seventh International Conference on Grid and Cooperative Computing, pp.185-189.

[10] Hesam Izakian, Ajith Abraham, and, Vaclav Snasel. "Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments",IEEE.

[11] T.Braun, H.Siegel, N.Beck, L.Boloni, M.Maheshwaran, A.Reuther, J.Robertson, M.Theys, B.Yao, D.Hensgen, and R.Freund, "A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems", In 8 th IEEE Heterogeneous Computing Workshop(HCW'99), pp. 15-29, 1999.

[12] Tracy D.Braun, Howard Jay Siegel, and Noah Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, pp.810-837, 2001.

[13] Qinma Kang, Hong He, Hongrun Wang, and Changjun Jiang, 1996. "A Novel Discrete Particle Swarm Optimization Algorithm for Job Schedulingin Grids", in IEEE Fourth International Conference on Natural Computation, pp.401-405.

[14] G. K. Kamalam and V. Murali Bhaskaran, "New Enhanced Heuristic Min-Mean Scheduling Algorithm for Scheduling Meta-Tasks on Heterogeneous Grid Environment", in European Journal of Scientific Research, Vol.70 No.3 (2012), pp. 423-430.

[15] Yang, X.S., Nature-inspired metaheuristic algorithms. 2010: Luniver Press.

[16] Senthilnath, J., S. Omkar, and V. Mani, Clustering using firefly algorithm: Performance study. Swarm and Evolutionary Computation, 2011.

[17] Adil Yousif, Abdul Hanan Abdullah, Sulaiman Mohd Nor and Adil Ali Abdelziz, "SCHEDULING JOBS ON GRID COMPUTING USING FIREFLY ALGORITHM", in Journal of Theoretical and Applied Information Technology, 2011. Vol. 33 No.2.

[18] Ku Ruhana Ku-Mahamud and Husna Jamal Abdul Nasir. "Ant Colony Algorithm for Job Scheduling in Grid Computing", in 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, IEEE Computer Society 2010.

[19] S. Gomathi and Dr.D.Manimegalai, "An adaptive grouping based Job Scheduling in Grid Computing", in Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011), IEEE 2011.

[20] Y. Huang, A. Brocco, N. Bessis, P. Kuonen, B. Hirsbrunner, "Exploring Decentralized Dynamic Scheduling for Grids and Clouds using the Community-Aware Scheduling Algorithm", in: 2011 Future Generation Computer Systems, Elsevier.