An Effective Cluster Score Job Scheduling Algorithm for Grid Computing

Vinodhini N., Dr. Kamalam G.K.

Department of Information Technology, Kongu Engineering College, Perundurai, Erode-638052, India
Anna University, Chennai

vinoguru053@gmail.com, gkk@kongu.ac.in

Abstract. *Grid computing is the collection of computer resources from multiple locations to reach a common goal. Distributed computing supports resource sharing. Parallel computing supports computing power. Grid computing aims to harness the power of both distributed computing and parallel computing. Grid can achieve the same level of computing power as a supercomputer does, also with a lowest cost. Grid is also a heterogeneous system. Grid comprises a collection of clusters. Scheduling independent task is more complicated in grid environment. In order to utilize the power of grid computing efficiently, an Adaptive Scoring Job Scheduling Algorithm (ASJS) assigns job to the resources by calculating the cluster scores. Cluster score is defined as the weighted value of the Average Transmission Power and Average Computing Power of cluster. An ASJS Algorithm is used to reduce the completion time of the submitted job, which may compose of computing-intensive jobs and data-intensive jobs in a grid environment.*

Keywords

Grid Computing, Job Scheduling, Resource Allocation, Cluster Formation

1 Introduction

The computational capability and network performance have gone to a great extent, there are still problems in the fields of science, engineering, and business, which cannot be effectively dealt by using the current generation of supercomputers. The emergence of the Internet as well as the availability of powerful computers and high-speed network technologies as low-cost commodity components is rapidly changing the computing landscape and society. These technology opportunities have led to the possibility of using wide-area distributed computers for solving large-scale problems, leading to what is popularly known as Grid computing.

Grid computing is the collection of computer resources from multiple locations to reach a common goal. Grid can achieve the same level of computing power as a supercomputer does, but at a much reduced cost. Grid is like a virtual supercomputer. Distributed computing supports resource sharing. Parallel computing supports computing power. Grid computing aims to harness the power of both distributed computing and parallel computing. The goal of grid computing is to aggregate idle resources on the Internet such as Central Processing Unit (CPU) cycles and storage spaces to facilitate utilization.

Grid computing enables the sharing, selection and aggregation of a wide variety of geographically distributed resources including supercomputers, databases, data sources and specialized devices owned by different organizations. The Grid users need not be aware of the computational resources that are used for executing their applications and storing their data.

2 Problem definition

Grid environment is heterogeneous in nature. Jobs arriving the grid are heterogeneous in nature. The scheduling of the jobs to the appropriate resources in a heterogeneous environment is complicated. Grid comprises of ‘n’ number of resources and ‘m’ number of jobs to be scheduled. The scheduling problem consists of mapping ‘m’ jobs to the ‘n’ resources.

3 Related work

Saha et al (1995) proposed the Fastest Processor to Largest Task First Scheduling Algorithm (FPLTF) is a good representative for Bag-of-Tasks applications. The strategy of the FPLTF scheduling algorithm is to schedule jobs according to the workload of jobs and computing power of resources.

Maheswaran et al (1999) proposed the Min–min scheduling algorithm, each job will be always assigned to the resource which can complete it earliest in order to spend less time completing all jobs. The Max–min scheduling algorithm is similar to Min–min scheduling algorithm. It gives the highest priority to the job with the maximum earliest completion time. Fairness is the key idea of the Round Robin (RR) scheduling algorithm.

Maheswaran at al (1999) proposed the On-line mode heuristic scheduling algorithms. Jobs are scheduled as soon as it arrives. Because a grid environment is heterogeneous with different types of resources, on-line mode heuristic scheduling algorithms are more appropriate for grid environment. Dynamic FPLTF Scheduling Algorithm (DFPLTF)is based on FPLTF scheduling algorithm and is modified to make the FPLTF scheduling algorithm more adaptive for grid environment.

Xu et al (2003)proposed the simple grid simulation architecture and modified the basic ant algorithm for job scheduling in grid. The scheduling algorithm they proposed needs some information such as the number of CPUs, Million Instructions Per Second(MIPS) of every CPU for job scheduling. A resource must submit the information mentioned above to the resource monitor.

Sheng-De Wang et al(2005) proposed the Most Fit Task First Scheduling Algorithm (MFTF) mainly attempts to assign the most suitable resource to the task by a value called fitness.

Abraham Silberschatz et al (2011) proposed the Batch mode heuristic scheduling algorithms. In that Jobs are queued and collected into a set when they arrive in the batch mode. They will be scheduled afterwards by the scheduling algorithm. Batch mode heuristic scheduling algorithms are more appropriate for the environment with the same type of resources. First-Come, First- Served Scheduling Algorithm (FCFS) is the simplest algorithm for job scheduling. Jobs are executed according to the sequence of job submitting. The second job will be executed when the first job is done, and therefore FCFS has a serious problem called convoy effect. The convoy effect will happen when there is a job with large workload in the front of the job sequence. All other small workload jobs have to wait until the big one finishes.

4 Proposed system

First initialize GridSim Toolkit with n number of users and the router is initialized. The resource is registered in GIS. Create n number of resources with n number of machines. 'n' number of resources will be created and ID will be randomly generated for each resource. Each resource contain 'n' number of machines and each machine contain random number of PEs(Processing Elements) and MIPS(Million Instruction Per Second) is assigned for each PE. Cost and baud_rate is assigned to each PE. Job is called as Gridlet. 'n' number of Gridlet will be created and ID will randomly generated to each job. Length of the job is assigned.

Initialize the load of each resource. The user submit computing-intensive jobs and data-intensive jobs. The computing-intensive job means that jobs need lots of computing power to complete and the data-intensive job means that the resource needs to take lots of bandwidth to transmit files.

4.1 Computing power

$$CP_k = CPU\_Speed_k * (1 - load_k) \quad (1)$$

CP_k – Available Computing Power of resource k.

4.2 Average transmitting power

ATP_i means the average available bandwidth the cluster i can supply to the job and is defined as:

$$ATP_i = \frac{\sum_{j=1}^{m} Bandwidth\_available_{ij}}{m-1}, i \neq j \quad (2)$$

$Bandwidth\_available_{i,j}$ is the available bandwidth between cluster i and cluster j, m is the number of clusters in the entire grid system.

4.3 Average computing power

ACP_i means the average available CPU power cluster i can supply to the job and is defined as:

$$ACP_i = \frac{\sum_{k=1}^{n} CPU\_Speed_k * (1 - load_k)}{n}, \text{ i = 0 to 2} \quad (3)$$

CPU_Speed_k is the CPU speed of resource k in cluster i, $load_k$ is the current load of the resource k in cluster i, n is the number of resources in cluster i.

4.4 Cluster score

The cluster score is defined as:

$$CS_i = \alpha.ATP_i + \beta.ACP_i, \text{ i = 0 to 2} \quad (4)$$

CS_i is the cluster score for cluster i, α and β are the weight value of ATP_i and ACP_i respectively, the sum of a and b is 1, ATP_i and ACP_i are the average transmission power and average computing power of cluster i respectively.
For a Grid environment the user might require a job to be cost effective or time effective. For a time effective job, the resource with lowest time in the chosen cluster is selected. For a cost effective job, the resource with lowest cost in the chosen cluster is selected. This is repeated for n number of Gridlets. Finally Makespan and processing cost is calculated.

4.5 Scheduling jobs to cluster

After calculating the cluster score of each cluster, the Job Scheduler will select the resource with the best computing power in the cluster with the highest cluster score and assign the job to the resource. For a Grid environment the user might require a job to be cost effective or time effective. Based on the type of job, it is submitted.

After a resource receives a job, it starts to execute. The status of the resource will change, and therefore local update will be applied to adjust the cluster score of the cluster containing the resource. Once a job is completed by a resource, the result will be sent back and stored in the Information Server.

5 Result analysis

There are three clusters in the grid system and each cluster has three resources. The initial status of each resource is stored in Information Server. Grid user_0 Creating 10 Gridlets. Job0 arrives at Job Scheduler, Job Scheduler will identify the type of job1 as computing-intensive and set the value of α and β. The value of α and β are 0.3 and 0.7 respectively. Then Job Scheduler will calculate the ATP, ACP and CS of each cluster and select the resource with the best CP in the cluster with the highest CS to execute job0. According to the cluster score of each cluster, job0 will be submitted to resource 0 in cluster 0. Local update will be executed and the load of resource 0 becomes 35%.

Table: Resource and Job Characteristics

Cluster	Resource ID	CPU_Speed (MHZ)	Load (%)
0	0	2796	30.0
	3	2661	31.0
	6	2652	26.0
1	1	2835	33.0
	2	3041	31.0
	8	3060	28.0
2	4	3191	25.0
	5	3146	32.0
	7	3205	32.0

After Job Scheduler submits job0, it continues to schedule job1, because job2 is data-intensive, the value of α and β set by Job Scheduler are 0.7 and 0.3 respectively. According to the cluster score of each cluster, job1 will be submitted to resource 3 in cluster 0. Local update will be executed and the load of resource 2 becomes 31%. Job0 is completed by resource 0 after Job Scheduler submits job1. The global update will be executed. Job2 will be scheduled by the Job Scheduler similarly.

For job 0 Minimum processing time 15.1 in Resource 0. This is repeated for n number of Gridlets. Finally Makespan and processing cost is calculated.

Table: Calculating Time and Cost

Resource	Time (sec)	Cost ($)
0	15.1	45.3
3	16.5	16.5
6	16.8	151.200

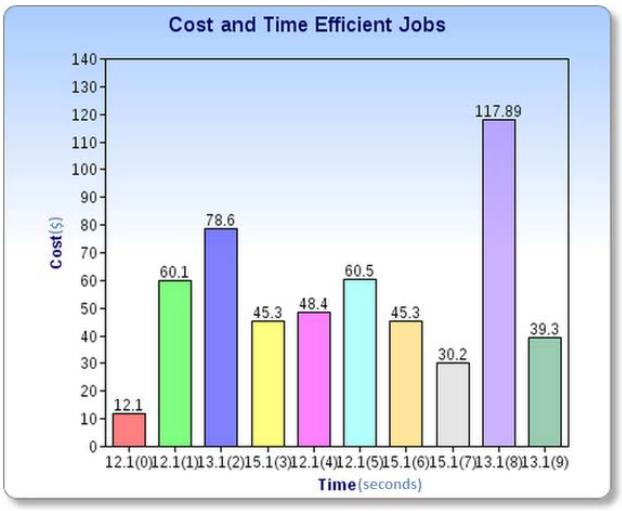

Figure: Cost and Time Efficient Jobs

From this graph job0 having time and cost efficient job among all these obtained jobs.

6 Conclusion

An cluster scoring method is to schedule jobs in grid environment. ASJS selects the fittest resource to execute a job according to the status of resources. Local and global update rules are applied to get the newest status of each resource. Local update rule updates the status of the resource and cluster which are selected to execute the job after assigning the job and the Job Scheduler uses the newest information to assign the next job. Global update rule updates the status of each resource and cluster after a job is completed by a resource. It supplies the Job Scheduler the newest information of all resources and clusters such that the Job Scheduler can select the fittest resource for the next job. The experimental results show that ASJS is capable of decreasing completion time of jobs and cost.
In future, ASJS will be implemented in real business grid applications. ASJS algorithm focuses on job scheduling. Modified ASJS shall be considered for division of file and the replica strategy in data-intensive jobs.

References

1. Maheswaran.M, Ali.s, Siegel.H.J, Hensgen.D, Freund.R(1999), 'Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing system', Journal of Parallel and Distributed Computing, pp. 107–131.
2. Ruay-Shiung Chang, Chih-Yuan Lin, Chun-Fu Lin (2012), 'An Adaptive Scoring Job Scheduling algorithm for grid computing', Journal of Information Sciences, pp. 79–89.
3. Saha.D, Menasce.D, Porto.S(1995), 'Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures', Journal of Parallel and Distributed Computing, pp. 1–18.
4. Sheng-De Wang, I-Tar Hsu, Zheng-Yi Huang(2005), 'Dynamic scheduling methods for computational grid environment', International Conference on Parallel
and Distributed Systems, pp. 22–28.
5. Xiaoxia Zhang, Lixin Txang(2005), 'CT-ACO – hybridizing ant colony optimization with cycle transfer search for the vehicle routing problem', Congress on Computational Intelligence Methods and Applications, p. 6.
6. Yun-Han Lee, Seiven Leu, Ruay-Shiung Chang (2011), 'Improving job scheduling algorithms in a grid environment', Journal of Future Generation Computer Systems, pp. 991–998.