An Analysis of MIPS Group Based Job Scheduling Algorithm with other Algorithms in Grid Computing

S. Gomathi,
A.P,FXEC,
Tirunelveli,
Tamilnadu,India
Mobile: 9944866629

Dr.D.Manimegalai,
Prof & Head, IT Dept,NEC,
Kovilpatti,
Tamilnadu,India
Mobile: 9442636698

Abstract

Two major problems in grid computing applications are, resource management and job scheduling. These problems do occur due to distributed and heterogeneous nature of the resources. This paper introduces a model in job scheduling in grid computing environments. A dynamic scheduling algorithm is proposed to maximize the resource utilization and minimize processing time of the jobs. The proposed algorithm is based on job grouping. The results show that the proposed scheduling algorithm efficiently utilizes resources at its best and reduces the processing time of jobs.

Keywords- Grid computing; Job grouping; Job scheduling; Dynamic scheduling; First come first served (FCFS) algorithm.

1. INTRODUCTION

Grid computing refers to the cooperation of multiple processors and its aim is to use the computational power in the areas which need high capacity of the CPU. The Grid is concerned with the exchange of computer power, data storage, and access to large databases, without users searching for these resources manually. Grid computing is based on large scale resources sharing in an Internet. Computational Grids are emerging as a new computing paradigm for solving challenging applications in science, engineering, economics and econometrics [1]. Computational Grid can be defined as large-scale high-performance distributed computing environments that provide access to high-end computational resources. And also it is defined as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, performance, capability, cost, and user's quality-of-service requirements.

Grid scheduling is the process of scheduling jobs over grid resources. A grid scheduler is in-charge of resource discovery, grid scheduling (resource allocation and job scheduling) and job execution management over multiple administrative domains. In heterogeneous grid environment with its multitude of resources, a proper scheduling and efficient load balancing across the grid can lead to improved overall system performance and a lower turn-around time for individual jobs. There are two types of scheduling namely static scheduling and dynamic scheduling in grid computing system. For static scheduling, jobs are assigned to suitable resources before their execution begin. For the dynamic scheduling, reevaluation is assigned to already taken assignment decisions during job execution.

In grid computing system, resources are not under the central control and can enter and leave the grid environment at any time. An effective grid resource management with good job and resource scheduling algorithm is needed to manage the grid computing system. In grid computing environment, there exists more than one resource to process jobs. One of the main challenges is to find the best or optimal resources to process a particular job in term of minimizing the job computational time. Optimal resources refer to resources having high CPU speeds and large memory spaces. Computational time is a measure of how long that resource takes to complete the job.

In a Grid computing environment, the scheduler is responsible for selecting the best suitable machines or computing resources for processing jobs to achieve high system throughput [2]. The scheduler must use coarse-grained jobs instead of light weight jobs so as to reduce communication and processing time. This paper focuses on grouping based job scheduling and how they are grouped as coarse grained jobs. The grouped jobs are allocated to resources in dynamic grid environment taking into account memory constraint, processing capabilities ,and the bandwidth of the resources.

This paper is organized as follows. In Section II, related work is surveyed, in section III basic grouping based job scheduling model is discussed, in section IV analyses experimental evaluation using GridSim toolkit [6] and section V concludes the paper with future work.

2. RELATED WORK

In the field of grid resource management and job scheduling, researchers have done much valuable work. Various algorithms have been proposed in recent years and each one has particular features and capabilities. In this section we review several scheduling algorithms which have been proposed in grid environment. Jobs submitted to a grid computing system need to be processed by the available resources. Best resources in terms of processing speed, memory and availability status are more likely to be selected for the submitted jobs during the scheduling process. Best resources are categorized as optimal resources.

A scheduling optimization method should consider the following two aspects, one is the application characteristics, and the other is the resource characteristics [6]. Taking the characteristics of lightweight job, into account there are some researches on the fine-grained job scheduling problem.

A dynamic job grouping-based scheduling algorithm groups the jobs according to MIPS of the available resources. This model reduces the processing and communication time of the job, but this algorithm doesn't take the dynamic resource characteristics into account and the grouping strategy may not utilize resource sufficiently [3].

A Bandwidth-Aware Job Grouping-Based scheduling strategy schedules the jobs according to the MIPS and bandwidth of the selected resource, and sends job group to the resource whose network bandwidth has highest communication or transmission rate. But, the strategy does not ensure that the resource having a sufficient bandwidth will be able to send the job group within required time [5].

Scheduling framework for Bandwidth-aware strategy schedules jobs in grid systems by taking of their computational capabilities and the communication capabilities of the resource’s into consideration. It uses network bandwidth of resources to determine the priority of each resource. The job grouping approach is used in the framework where the scheduler retrieves information of the resources processing capability. The scheduler selects the first resource and groups independent fine-grained jobs together based on chosen resources processing capability. These jobs are grouped in such a way that maximizes the utilization of the resource’s and reduces the total processing time. After grouping, all the jobs are sent to the corresponding resource’s whose connection can be finished earlier which implies that the smallest request is issued through the fastest connection giving best transmission rate or bandwidth. However, this strategy does not take dynamic characteristics of the resources into account, and preprocessing time of job grouping and resource selection are also high [4].

The above analysis of various grouping based job scheduling strategy presents some of their advantages and disadvantages .However, there are some defects in the above scheduling algorithms. First, the algorithms doesn’t take the dynamic resource characteristics into account. Second, the grouping strategy can’t utilize resource sufficiently. And finally, it doesn’t pay attention to the network bandwidth and memory size. To solve the problems mentioned above, an adaptive fine grained job scheduling mechanism is presented in this paper.

3. JOB SCHEDULING MECHANISM

The job scheduler is a service that resides in a user machine. Therefore, when the user creates a list of jobs in the user machine, these jobs are sent to the job scheduler for scheduling arrangement. The job scheduler obtains information about the available resources from the Grid Information Service (GIS). Based on this information, the job scheduling algorithm is used to determine the job grouping and resource selection for grouped jobs. The size of a grouped job depends on the processing requirement length expressed in Million Instructions, Bandwidth expressed in MHz/s and Memory size requirement expressed in MB, expected execution time in seconds. As soon as the jobs are put into a group with a matching selected resource, the grouped job is dispatched to the selected resource for computation.

The grouping strategy should be based on the characteristics of resources. In grid computing, there are two approaches for obtaining dynamic resource characteristics for job execution. One is that a user directly searches the resources for job execution using an information service. The other is to use a resource manager. With a resource manager, users can obtain information about the grid through an interactive set of services, which consists of an information service that is responsible for providing information about the current availability and capability of resources. The resource monitoring mechanism used in the proposed algorithm belongs to the second one.

Grouping strategy is done based on the resource's status according to processing capabilities (in MIPS), bandwidth (in MHz/s), and memory size (in MB) of the available resources. After gathering the details of user jobs and the available resources, the system selects jobs in FCFS order to form different job groups. The scheduler selects resources in FCFS order after sorting them in descending order of their MIPS. Jobs are put into a job group one after another until sum of the resource requirements of the jobs in that group is less than or equal to the amount of resources available at the selected resource site. Here, only the processing capability and bandwidth are used to constrain the sizes of coarse-grained jobs, but we can easily join additional constraints .Then the fine-grained jobs can be grouped as several new jobs and these new jobs should satisfy the following formula:

1. $MI(job\_group\_i) \leq MIPS(i)*tp(i)$
2. $FSG(job\_group\_i) \leq BW(i)*tc(i)$
3. $TMR(all\_group) \leq TMA(all\_resources)$
4. $tp>tc$

In the above conditions, MI(job_group_i) is the processing capacity of the resource i which will be allocated to the jobgroupi , tp(i) is the expected job processing time, FSG(job_group_i) is the file_size (in Mb) of the jobgroupi at the resource i , tc(i) is the communication time, BW(i) is the bandwidth of resource i, TMR denotes the total amount of memory needed during the execution of the job j, TMA denotes the total amount of memory available.

Equation (1) specifies that the processing time of the coarse-grained job shouldn't exceed the expected time. The communication time of the grouped jobs should not exceed computation time of the grouped jobs and this is illustrated as (2) & (4). Equation (3) specifies that the memory size requirement of the jobgroup shouldn't exceed to the resource memory size. These are the constraints in job grouping.

This algorithm is divided into two parts. In the first part, the scheduler receives resource status using GIS. And, it sorts job list in descending order, and assigns a new ID for each job. In the second part after gathering the details of user jobs and the available resources, the system selects Jobs in FCFS order to form different job groups. The scheduler selects resources in FCFS order after sorting them in descending order of their MIPS. Jobs are put into a job group one after another until sum of the resource requirements of the jobs in that group is less than or equal to amount of resource available at the selected resource site.

In this way jobs are subsequently gathered or grouped one by one according to the resulting MIPS, Memory size and Bandwidth of the resource until the above conditions are satisfied. As soon as a job group is formed, the scheduler submits the grouped job to the corresponding resource for job computation setting the resource power to zero. After execution the job group, the results goes to the corresponding users and resource is again available to Grid system.

Algorithm:

Begin

Part 1: Initialization

Step 1.

Direct jobs to the Scheduler.

Step 2.

```
Direct Resource status to the scheduler
Step 3.
Sort joblist in descending order based on MIPS
Part 2: Job Scheduling
Step 1.
[Traverse Joblist]  For i<-0 to joblistsize-1 do through step 2
Step 2.
[Traverse Grouplist] For j<-0 to joblistsize-1 do through step 3
Step 3
[Compare Processing Time]  if MI(job_group_i)+jobi  ≤
MIPS(i)*tp(i)
And
[Compare File Size] (jobgroup_ file_ sizej + job_ file_   sizei ) /
baud_ ratej <tp(i) ) or MI( jobgroup_j )  = 0,
And
[CompareMemory] jobi/MIPSj > job_file_size  /baud_ratej)
Step 4 [Construct job group] add job i to job group j;
Step 5 [Loop Break] break;
Step 6 [End Compare] endif
Step 7 [Increment j]   j++;
Step 8 [End Loop] endfor
Step 9 [Compare Status of i]  if job i can't join any job_group then
Step 10 [Construct joblist2]  add job i to joblist2;
Step 11 [Compare End] endif
Step 12 [Increment i] i++;
Step 13 [End Loop] endfor
Step14[Travese jobgrouplist_size]  for i:<-0 to jobgrouplist_size-1
Step 15[Allocate jobgroup to resource]  jobgroup I <- resource i;
Step 16[End Loop] endfor
Step 17[Compare size of joblist2]  if joblist2_size<>0 then
Step 18[Assign joblist] joblist<-joblist2;
Step 19[Synchronize Process] wait a while;
Step 20[Get Resource status] get resource status from GIS;
Step 21[Receive jobgroup]  receive computed jobgroup from
resources;
Step 22 [Loop part2] repeat part2;
Step 23[Compare End] endif
Step24 [Receive Computed Jobgroup] receive computed job group
from resources.
Step 25 [End of Algorithm] Ends
```

There are disadvantages in the above discussed algorithm. One of the major disadvantages is that there are some specific set of jobs that require only to a specific set of resources for assignment. For an instance, the job jx can be done only by the resource rx. Therefore the job jx cannot be assigned to any other resource. Another disadvantage is that as it is also possible that some job may require the processing capabilities of more than a resource. Either a parallel assignment or sequential assignment may be considered as a solution at times. In grid computing architecture dynamic scheduling, the resources are not at all under central control. A resource may enter and leave the environment at any time. The frequency of how frequently a resource enters to grid environment stays and moves away have to be carefully accounted. This statistics will help in placing the jobs in queue. The resource that most frequently enters into the grid can handle processing of jobs without much delay. A random based assignment could be a most ideal choice in the scheduling structure. Resource Manager gets information about the next entering resource into the grid environment. This in turn informs the job scheduler to regroup jobs that can be assigned to the entering resource. The job scheduler informs the list processor to regroup the jobs that can be assigned to the incoming resource. The group of jobs thus rescheduled will be made available to the resource. If there are n numbers of resources that a processor r can process, then the jobs within the group can be diverted to the resource in the FCFS pattern. The factors such as bandwidth, processing capabilities and memory size should be accounted in listing the jobs within a group (ordering). Jobs that require resource processing may be put on cycle till it finds a suitable resource entering into the grid.

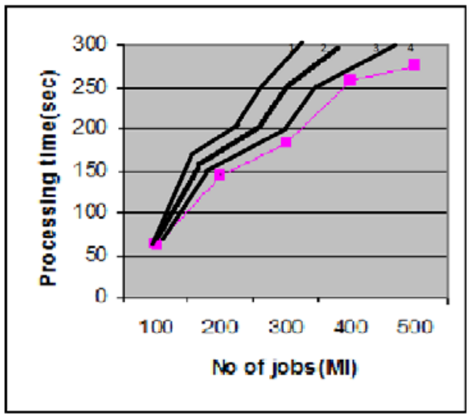
4. EXPERIMENTAL EVALUATION

GridSim [6] has been used to create the simulation of grid computing environment. In this simulation, each resource is characterized by its MIPS, bandwidth and memory size. The jobs are characterized by their amount of computations, expected execution time, memory-size requirement and expected transfer time. In this experiment, jobs and resources are randomly generated and the number of jobs varies from 100 to 500. Jobs in different groups are given different amount of execution time. The processing time is taken into account to analyze the feasibility of the proposed scheduling algorithm. Our algorithm can reduce the execution time and also the job completion success rate is high.

Table1: Job processing table

SNO	No.of Jobs	Processing Time(FCFS)	Processing Time(GBDJS)
1	100	55	55
2	200	200	160
3	300	280	220
4	400	380	260
5	500	440	280

Figure1.Job Processing Time.



Note: Graph numbered 1 shows the behavior of FCFS Algorithm whereas Line segment number 3 shows the performance of Ant Colony optimization Algorithm. The deviation shows that it consumes comparatively lesser time than that of FCFS because of grouping strategy applied on it. The disadvantage with the algorithm is it has not taken care of other parameters such as bandwidth, etc. Graph numbered 2 still shows better performance than that of the previous two because of the application both grouping and priority in the jobs in the group. The sum of all priorities of the group is accounted as priority of the group. One problem associated with this algorithm is how to handle the tie of two or more groups having same priority. This conflict can be overcome by introducing priority resolver. Though it is effective, it takes account of only jobs of similar nature. Line number 4 shows the behavior of grouping based dynamic job scheduling algorithm. Performance of grouping based job scheduling algorithm consumes lesser processing time in comparison with the all other algorithms accounted for because of the following facts.

Case 1: MIPS of job is much lesser than the MIPS of resource. The resource is not fully utilized and no further assignment is done till the MIPS of resource expires.

Case 2: MIPS of job equals to the MIPS of resource. Here the resource is fully utilized.

Case 3: MIPS of job is greater than the MIPS of resource. This assignment will not work because of the lesser MIPS of the resource. A couple of strategies is suggested. The first one is to discard the resource and wait in the queue till the job finds a suitable resource in terms of MIPS and carry out scheduling accordingly. The second option is to carry out the process partly and the part of unfinished process can be assigned to another resource subsequently.

5. CONCLUSION

In order to utilize grid resources efficiently, an adaptive fine grained job scheduling algorithm is proposed. The proposed Scheduling Model in Grid Computing is a grouping based job scheduling strategy that has taken memory constraint of individual jobs together with expected execution time at the job level into account rather than at the group level. The grouping algorithm improves the processing time of fine grained jobs. Experimental result demonstrates efficiency and effectiveness of the proposed algorithm. Though the proposed algorithm can reduce the execution time, its time complexity is high and some improvement should be done in this aspect. The proposed model reduces the waiting time of the grouped jobs. To further test and improve the algorithm, some dynamic factors such as high priority, network delay and QoS constraints can be taken into account. Advantages of this algorithm compared with others are: It reduces the total processing time of jobs. It maximizes the utilization of the resource. Minimizing the wastage of CPU power. Grouping the jobs fine-grained into grouping coarse grained will reduce the network latencies.

REFERENCES

[1] Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure.Morgan Kaufmann (1998)

[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, 2002,pp. 1175–1220.

[3] N. Muthuvelu, Junyan Liu, N.L.Soe, S.venugopal, A.Sulistio, and R.Buyya "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," in Proc of Australasian workshop on grid computing, vol. 4, 2005,pp. 41–48.

[4] Ng Wai Keat, Ang Tan Fong, "Scheduling Framework For Bandwidth-Aware Job Grouping-Based Scheduling In Grid Computing", Malaysian Journal of Computer Science, vol.19, No. 2, 2006,pp. 117-126 .

[5] T.F. Ang, W.K. Ng, "A Bandwidth-Aware Job Scheduling Based Scheduling on Grid Computing", Asian Network for Scientific Information, vol. 8, No. 3, pp. 372-277, 2009.

[6] V. Korkhov, T. Moscicki, and V.Krzhizhanovskaya, "Dynamic workload balancing of parallel applications with user-level scheduling on the grid," Future Generation Computer Systems, vol.25, January 2009, pp.28-34,

[7] F. Dong and S. G. Akl, "Scheduling algorithm for grid computing: state of the art and open problems," Technical Report of the Open Issues in Grid Scheduling Workshop, School of Computing, University Kingston, Ontario, January 2006.

[8] Quan Liu, Yeqing Liao, "Grouping-based Fine-grained Job Scheduling in Grid Computing", IEEE First International Workshop on Educational technology And Computer Science,vol.1, 2009, pp. 556-559.

[9] Dr. G. Sudha Sadasivam, "An Efficient Approach to Task Scheduling in Computational Grids", International Journal of Computer Science and Application, vol. 6, No. 1, 2009, pp. 53-69.

[10] K.Somasundaram, S.Radhakrishnan, "Node Allocation In Grid Computing Using Optimal Resource Constraint (ORC) Scheduling",IJCSNS International Journal of Computer Science and Network Security, vol.8 No.6, June 2008.

[11] C. Liu, and S. Baskiyar, "A general distributed scalable grid scheduler for independent tasks," J. Parallel and Distributed Computing, vol. 69, no. 3, 2009 , pp. 307-314 .

[12] Nikolaos D. Doulamis, Emmanouel A. Varvarigos ," Fair Scheduling Algorithms in Grids" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 11, NOVEMBER 2007,pp. 1630- 1648.

[13] Y. C. Liang and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," IEEE Trans. Reliability,vol. 53, no. 3,2004, pp. 417–423.

[14] Vishnu Kant Soni, Raksha Sharma, Manoj Kumar Mishra ," An Analysis of Various Job Scheduling Strategies in Grid Computing " , 2nd International Conference on Signal Processing Systems (ICSPS),2010 , pp.162-166.