Job Scheduling Model with Job Sequencing and prioritizing strategy in Grid Computing

A.R. Jayasudha
Assistant Professor -Department of Computer Applications,
Hindusthan College of Engineering and Technology,
Coimbatore-641032, Tamilnadu, India

T. Purusothaman, PhD.
Associate Professor - Department of Computer Science,
Government College of Technology,
Coimbatore-641012, Tamilnadu, India

ABSTRACT
In a heterogeneous distributed systems that involves parallel processing it is very significant to include its computational capability and its dynamism as a major consideration. In such a distributed and a dynamic environment, the application and usage of available resources vary in a higher magnitude. In order to perform maximum exploitation of all available resources for parallel processing, efficient and intellectual scheduling is the need of the hour. Current grid environment regards clusters as separate domains. As resources are highly heterogeneous, dynamic and are separated largely by geographical distances, efficient job scheduling is a challenge. In this paper, we have discussed about a scheduling model that performs job sequencing based on shortest processing time and earliest due date. The simulation results show a considerable improvement in the resource utilization, throughput and reduction in waiting time as compared to others.

General Terms
Grid Computing, Tasks, Resources, Scheduling.

Keywords
Resource Scheduling, Job Sequencing, Priority

1. INTRODUCTION
In a grid environment, distributed resources belong to different administrative domains[1]. Execution of jobs is performed across organizations for many different applications. Grid technologies complement rather than compete with existing distributed computing technologies [2]. The Globus Project defines Grid as "an infrastructure that enables the integrated, collaborative use of high-end computers, networks, databases and scientific instruments owned and managed by multiple organizations[4]. In general, grid users urge to run additional jobs and expect immediate and fast results. With more tasks an d more number of machines, an efficient scheduler is required which can interact with the cluster that is locally available to schedule jobs though each cluster follow different and own scheduling policy.

2. PREVIOUS RESEARCH
In order to gain maximum benefit Guarantee of victorious probability algorithm(GVP) based on game theory was found. It predicted the results based on historical information. An in-depth analysis was made using two-player game and multi-player game[3]. In [5] usefulness of scheduling a task is discussed. Global and local clusters are used for dynamic scheduling.

Long term and short term decisions were based on these global and local clusters[6].The scalability of ant algorithm was validated and a grid simulation architecture was formulated.[7]. In [8], with reference to the Grads architecture, a meatascheduler was discussed.

The objective was to minimize the task execution time by stopping the bigger job and letting the smaller jobs to proceed. Two level time sharing scheduling for simultaneous and sequential jobs were introduced where in time slots were divided into two for both the types of jobs.

In [10], a scheduling technique that predicts variance for scheduling decisions is discussed and the algorithm was adjusted based on the received feedback. Cluster Scheduling using game theory is discussed. The scheduling gave much preference to jobs from local site and least priority for jobs from remote site[11].

Local cluster policies have strong influence on response time of jobs. A greedy approach was proposed to overcome the drawback[12]. The approach gave significant results but failed to throw light on the differences between the local site policies and remote site policies.

Multiple types of resources were regarded using a constraint language for resource selection[13].

However, there are a few deficiencies in the scheduling algorithms. The Proposed Model Job sequencing and prioritizing strategy in Computing aims at improving and increasing the throughput and aims at reducing the waiting time of the jobs that needs a very little time for execution.

3. SCHEDULING MODEL
A grid is an integrated package of hardware and software infrastructure providing consistent and economically conducive access to high power computational resources. Grid Scheduling involves users, global and cluster level[14]. When users submit their jobs, the schedulers at the global level receives the jobs and forma a queue. These queued jobs are sent for execution depending on the computational capability of resources at the cluster. Resources are owned by different domains and are shared based on its local policies.

Grid scheduling involves resource discovery, filtering and resource selection. Grid scheduler does not have direct control over the grid resources. An efficient and a proper task scheduling is possible only if there is adequate information about the available resources. Grid Information Service plays a major role in providing such information to grid schedulers. The global scheduler allocates the user jobs to local cluster according to their computational capability. The local scheduler has a pivot role to play in determining the overall performance of the system.

Local Resource Manager is responsible for local scheduling, where jobs from grid users as well as jobs from local domains are executed and updates Grid Information Service about the status. The scheduling decisions made by the global schedulers are changed by the local scheduler to adapt to changes in the availability of resources. During the execution of the scheduling process, the clusters are dynamically selected which aids in balancing the load if any particular cluster is overloaded. At the local cluster level, the jobs are executed and the results are sent back to the user. The global and local scheduler interacts with each other to make an optimal scheduling of jobs[14].

4. JOB SEQUENCING AND PRIORITIZING ALGORITHM

The sequencing problem involves the determination of an optimal order or sequence of performing a series of jobs to optimize the total time or cost. Sequencing aids in minimizing the mean flow time. Let there be n jobs, each of which has to be processed one at a time on m different machines. The actual processing time of each job on the machines is known.

$M_{ij} \rightarrow$ processing time for job i on machine j.

T $\rightarrow$ total elapsed time for processing the jobs.

$W_i \rightarrow$ Waiting time for job i

$F_i \rightarrow$ flow time of job i $F_i = W_i + t_i$

4.1. Assumptions

- Only one operation is carried out on a machine at a particular time.
- Each operation, once started must be completed.
- Jobs are processed only in the order specified.
-

5. OPTIMAL SEQUENCE ALGORITHM

5.1. Job Monitoring and Prioritizing Algorithm [JMPA]

Varying results have been accepted based on research done on distributed computing. But, dynamically growing and changing demands of the nature of jobs and requirements calls for new scheduling algorithms. A significant factor to be considered while scheduling jobs is its time delay. It is very cumbersome to estimate the execution time of any job as it depends on various parameters such as CPU capability, size of memory and Input / Output speed etc., As the dimension and depth of any engineering problem simulation is very different from each other, it is much harder to schedule them optimally.

The ultimate goal of the job monitoring and measuring algorithm is to minimize the waiting time of jobs. Jobs with shortest processing time are given highest priority by sequencing them accordingly which would automatically result in increased throughput. The JMPA takes into account the tentative processing time and generates the flow time and the average in-process inventory. Moreover, the algorithm takes the due date of the job completion also into account. The submitted jobs are sequenced using selection sort algorithm in different ways. In the first case, the jobs are sequenced based on their processing time. In the next case, it is sequenced based on the due date of the jobs. JMPA algorithm takes these sequenced arrays of jobs as input and processes. SPT (Shortest Processing Time) minimizes mean waiting time Wi and minimizes the flow time F. Eight jobs 1, 2… 8 are to be processed on 3 machines. The processing times, due dates of the jobs are represented in Table 1.1

Table 1: Jobs with processing time and due date

Job	Processing time ti (in minutes)	Completion times (in minutes)	Due date di
1	5	3	15
2	8	6	10
3	6	11	15
4	3	17	25
5	10	24	20
6	14	32	40
7	7	42	45
8	3	56	50

5.2. Selection Sort Algorithm.

Selection sort first finds the smallest in the array and exchanges it with the element in the first position, then find the second smallest element and exchanges it with the element in the second position, and continue in this way until the entire array is sorted.

Mean Flow time (MFT)

$\sum_{i=1}^{n}$ ji

$\sum_{i=1}^{n}$ Fi

As per SPT Rule, Optimal sequence is 4-8-1-3-7-2-5-6

Mean flow time=191/8 = 23.875 min.

Therefore, average in-process inventory is 191/56=3.41 jobs.

The part of the selection sort algorithm is as follows:

```
for (i=n down to 1)
   max = a[1];
   max_pos = 1;
            for (j=1 to i-1)
                if (a[j] > max)
                   max = a[j];
                   max_pos = j;
```

if (i != max_pos)

swap(i,max_pos);

5.3. Earliest Due Date Rule

By this rule, jobs are sequenced in the order of non-decreasing due dates. This minimizes the maximum job lateness as well as maximum job tardiness. As per EDD Rule,

Optimal sequence is 2-1-3-5-4-6-7-8

Mean flow time=256/8 = 32min.

Therefore, average in-process inventory is 256/56=4.57 jobs.

6. EXPERIMENTAL EVALUATION

GridSim has been used to create the simulation environment. A heterogeneous environment is used to conduct the simulation. The results show the improvement of the proposed model over the other scheduling models. In this testing, the total number of tasks and its average waiting time in the queue is taken into consideration to examine the feasibility of the algorithm.

6.1. Job Monitoring and Prioritizing Algorithm

Submit the user’s job to the grid.

Receive the job list.

Receive resource list.

//Perform JMPA

//Perform selection sort based on SPT

for(i=n down to 1)

max=a[1];

max_pos=1;

for(j=1 to i-1)

if(a[j]>max)

max=a[j];

max_pos=j;

if(i != max_pos)

swap(1,max_pos);

//Perform selection sort based on EDD

for(i=n down to 1)

max=a[1];

max_pos=1;

for(j=1 to i-1)

if(a[j]>max)

max=a[j];

max_pos=j;

if(i != max_pos)

swap(i_max,pos);

Divide the sorted jobs in both rules to halves.

for(i=0 to JobList / Number of Jobs do

if check <presence of Ji> in SP then

if check <presence if Ji> in EDD

Execute the Job

Endif

Endif

i++;

while JobList>0;

Co mplete more jobs with EDD and SPT

End;

Table 2: Comparison Based on Throughput

Job	Processing time t_i(in minutes)	Due date d_i(in minutes)
1	5	15
2	8	10
3	6	15
4	3	25
5	10	20
6	14	40
7	7	45
8	3	50

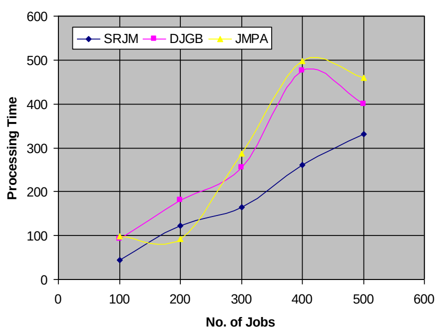

Figure 1: No. of throughputs

As compared to Secured Resource Job Scheduling Model and dynamic job grouping model, Job Monitoring and prioritizing algorithm shows improved results. By sequencing based on the shortest processing time, the result shows that there is maximum throughput. Smallest jobs that wait in the queue for an unnecessary long time has considerably been reduced, resulting in efficient allocation.

Table 3: Comparison Based on Waiting Time

No. of Jobs	SRJM	DJGB	JMPA
20	3	3	2
15	2	3	1
30	4	5	3
42	5	5	3
50	6	7	4

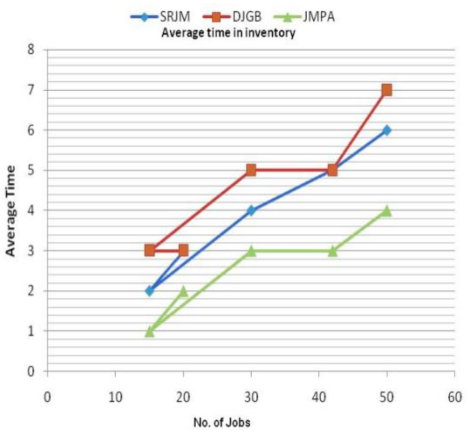

Figure 2: No. of Jobs in Inventory

7. CONCLUSION

Simulation is conducted to analyze and compare the throughput and waiting time between proposed algorithms JMPA and SRJM and DJGB algorithms. Result shows remarkable difference in the proposed algorithm, in which the numbers of jobs processed are more.

8. REFERENCES

[1] I. Foster, C. Kesselman, & S. Tuecke. The anatomy of the grid: enabling scalable virtual organizations, International

Journal of High Performance Computing Applications,15(3), 2001, 200-222.

[2] *The Anatomy of the Grid* www.globus.org/alliance/publications/papers/**anatomy**.pdf

[3] The 10th IEEE International Conference on High Performance Computing and Communications "A Novel Algorithm for Task Scheduling in Grid Computing Based on Game Theory" Lei Yao, Guanzhong Dai, Huixiang Zhang, Shuai Ren, Yun Niu College of Automatic, Northwestern Polytechnical University, Xi'an, China

[4] Rajkumar Buyya and Srikumar Venugopal - A Gentle Introduction to Grid Computing and Technologies - database, 2005 - gridbus.cs.mu.oz.au

[5] Seventh International Conference on Networking, "A price-based Task Scheduling for Grid Computing" Marcelo Massocco Cendron, Carlos Becker Westphall, Network and Management Laboratory, Federal University of Santa Catarina,Florianopolis, SC, Brazil

[6] K. Y. Wang, D. C. Marinescu and O. F. Carbynar, "Dynamic scheduling of process groups", Concurrency: Practice and Experience, 10(4):265-283, April 1998.

[7] "Ant Algorithm-Based Task Scheduling In Grid Computing", Zhihong XU Xiangdan HOU Jizhou SUN, Department of Computer Science, Tianjin University.

[8]. S. Vadhiyar, J. Dongarra, "A Metascheduler for the Grid", 11th IEEE international Symposium on High Performance Distributed Computing HPDC-2002.

[9]. B.Zhou, R. Brent, D.Walsh and K.Suzaki, "Job scheduling strategies for networks of workstations", In Job Scheduling Strategies for networks of workstations", vol.1459 of LNCS, pp.143-157, Berlin, 1998. Springer.

[10]. Fukuda Munehiro, Tanaka Yuichiro, Suzuki naoyha, Bic Lubomir F., Kobayashi Shinya, "A Mobile-Agent-Based PC Grid", In the 5th International Workshop on Active MiddlewareServices.

[11] Smith Warren, foster Ian, Taylor Valerie, "Scheduling with Advanced Reservations". In Proceedings of the 14th IEEE International Parallel and Distributed Processing Symposium2000.

[12] Streit Achim, "Evaluation of Scheduling-Algorithms for Metacomputing". Diploma thesis at CEI University of Dortmund, Germany (1999).

[13] Fernandez-Baca David., "Allocating Modules to processors in a distributed System" In IEEE Transactions on

Software Engineering, 15(11):1427-1436 (November 1989).

[14] "A Secure Resource and Job Scheduling Model with Job Grouping strategy in Grid Computing", Raksh Sharma, Vishnu Kant Soni, Manoj Lumar Mishra, Sarita Das School of Computer Engineering, Computer Science and Engineering KIIT, University, Est Phulnakhara, Orissa.