A Memory-Dependent Dynamic Job Scheduling Algorithm in Grid Computing

PRASHANTHA H A[1], SHIVARAJU R S[2]

[1]*Lecturer, Department of CS&E, Government Polytechnic, Devadurga, Karnataka*

[2]*Lecturer, Department of E&CE, Government Polytechnic, Devadurga, Karnataka*

***Abstract*— Grid computing is the ultimate framework that provides a high performance computing environment to meet growing and larger scale computational demands. However, Grid Computing is a critical and complex undertaking as the management of resources and computational jobs are geographically distributed under the ownership of different individuals or organizations with their own access policies, dynamic availability and heterogeneous in nature. Therefore, it is a big challenge and pivotal issue to design an efficient job scheduling algorithm for implementation in the real grid system. Various works has been done by many researchers, still further analysis and research needs to be done to design new techniques and improve the performance of scheduling algorithm in grid computing. The main purpose of this project is to develop an efficient job scheduling algorithm to maximize the resource utilization and minimize processing time of the jobs. The proposed job scheduling is based on job grouping concept taking into account Memory constraint together with other constraints such as Processing power, Bandwidth, expected execution and transfer time requirements of each job. These very constraints are taken at job level rather than at group level. The experimental results demonstrate that the proposed scheduling algorithm efficiently reduces the processing time of jobs in comparison to others.**

***Index Terms*:- Grid computing; Job grouping; Job scheduling.**

I. INTRODUCTION

The concept of grid computing is emerging as a new and important infrastructure for Internet-based high performance parallel and distributed computing. Grid computing, originally motivated by wide-area sharing of computational resources [1], has evolved to be mainstream technologies for enabling large-scale virtual organization [2]. he aspiration to share processing resources between many organizations to resolve large scale problems has provoked computational grids [3].The term "Grid" refers to systems and applications that integrate resources and services distributed across multiple control domains [4].Computational grids provide large-scale resource sharing, such as personal computers, clusters, MPPs, Data Base, and online instructions, which may be cross-domain, dynamic and heterogeneous [5]. To accomplish the sharing, exchange, discovery, and aggregation of resources distributed across multiple administrative domains, organizations and enterprises, Grid computing needs to support various services: security, uniform access, resource management, job scheduling, application composition, computational economy, and accounting. Among them, job scheduling is one of the important and difficult issues to address because it is NP Complete.

Although several initiatives have addressed the development of Grid scheduler, more attention is needed to design and enhance efficiency of job scheduling algorithm to reduce the processing time of jobs.

In a Grid computing environment, scheduler is responsible for selecting the best suitable machines or computing resources for processing jobs to achieve high system throughput [6]. The scheduler must use coarse-grained jobs instead of light weight jobs so as to reduce communication and processing time. This paper mainly focuses on grouping based job scheduling and how they are allocated to resources in dynamic grid environment taking into account memory constraint, expected execution and transfer time at the job level rather than at group level in grid system. This paper also focuses on memory requirements of the jobs at the time of execution along with other constraints such as processing power, and Bandwidth requirements of each job.

This paper is organized as follows. In Section II, related work is discussed in brief, Section III basic grouping based job scheduling model, Section IV memory-aware job scheduling model, Section V analyses experimental evaluation and VI gives conclusion and future work and lastly, the references.

II. RELATED WORK

An Agent Based Dynamic Resource Scheduling Model with FCFS-Job Grouping Strategy has of two major parts, first part provides resource management, which selects highest computational power cluster at the grid level and the second part provides FCFS-job grouping strategy that makes efficient utilization of the selected cluster by submitting a matching group of jobs from a FCFS job queue. This job grouping strategy minimizes total processing time and reduces the waiting time of the grouped jobs. The resource selection uses MHT algorithm to select the resource which takes O(log n) time. But this paper does not consider bandwidth and file size constraint except computational power of the resource [7].

Scheduling framework for Bandwidth-aware strategy schedules jobs in grid systems by taking into consideration of their computational capabilities and the communication capabilities of the resources. It uses network bottleneck 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 resources and reduces the total processing time. After grouping all the jobs sends to the corresponding resources whose connection can be finished earlier which implies that the smallest request issued through the fastest connection giving best transmission rate or bandwidth. However, this strategy does not take dynamic characteristics of the resources into account, preprocessing time of job grouping and resource selection is also high [8].

Grouping-based Fine-grained Job Scheduling model [9] is based on characteristics of resources. The fine-grained jobs grouped into forming coarse-grained are allocated to the available resources according to their processing capabilities in MIPS and bandwidth in Mb/s. The grouping algorithm integrates Greedy algorithm and FCFS algorithm to improve the processing of Fine-grained jobs. This model reduces the total processing time of jobs, maximizes the utilization of the resources and reduces the network latency. It does not pay any attention to memory size constraint and preprocessing time of job grouping is high. The time complexity scheduling algorithm is very high. 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 [6].

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 [10].

The above analysis of various grouping based job scheduling strategy presents some of their advantages and disadvantages and may prompt others to work further in improving and designing the scheduling strategy. No scheduling models discussed above have taken into account the memory constraint at resource site which is very important in reducing the processing time and successful execution of the jobs within the required time. In this paper processing and transfer time has been taken at the individual job level rather than at group level in contrast to other models discussed above, which is necessary and practical in realizing a real grid system. Therefore, the proposed "A Memory-Aware dynamic Job Scheduling model" is the outcome of this analysis.

III. BASIC GROUPING BASED JOB SCHEDULING MODEL

The job scheduler is a service that resides in a user machine figure 1. Therefore, when the user creates a list of gridlets or 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 Mb/s and Memory size requirement expressed in Mb, expected execution and transfer 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.

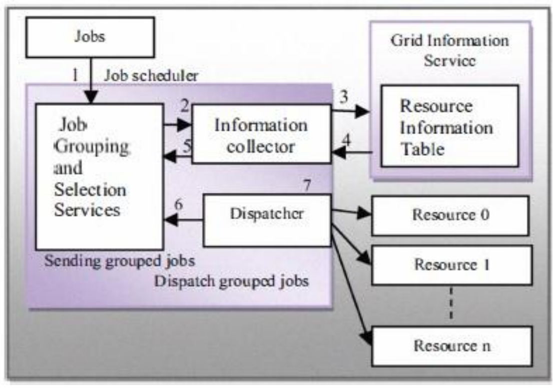

Figure 1. Grouping-based job scheduling model

IV. MEMORY-DEPENDENT JOB SCHEDULING MODEL

Memory dependent job scheduling model presents and evaluates an extension to Computational-Communication Memory size based job grouping scheduling strategy that tries to maximize the utilization of Grid resources and their processing capabilities, and also reduces processing time and network delay to schedule and execute the jobs on the Grid. The model groups the jobs according to jobs requirement and available resource capability.

A. Architecture of grouping based Job Scheduler

Figure 2 depicts the architecture of the job scheduler used in the system. All jobs accepted by the grid system over a certain period of time are specified by their Job_ID, Job_MI, Job_MemorySize, Expected Execution and Transfer Time. And during that period all available grid resources are specifIed by their Resource_ID, Resource_MIPS, Resource_Bandwidth, Resource_Memory Size in the grid environment (step 1-3). After gathering the details of user jobs and the available resources, the system selects Jobs in FCFS order to form different job groups (step 4). 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 (step 5). In this way jobs are subsequently gathered or grouped one by one according to the resulting MI, Memory size and Bandwidth of the resource until the condition on which it is based is satisfied (step 6). 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 (step 7). After execution the job group, the results goes to the corresponding users and resource is again available to Grid system with its available power and ready to execute another job (step 8).

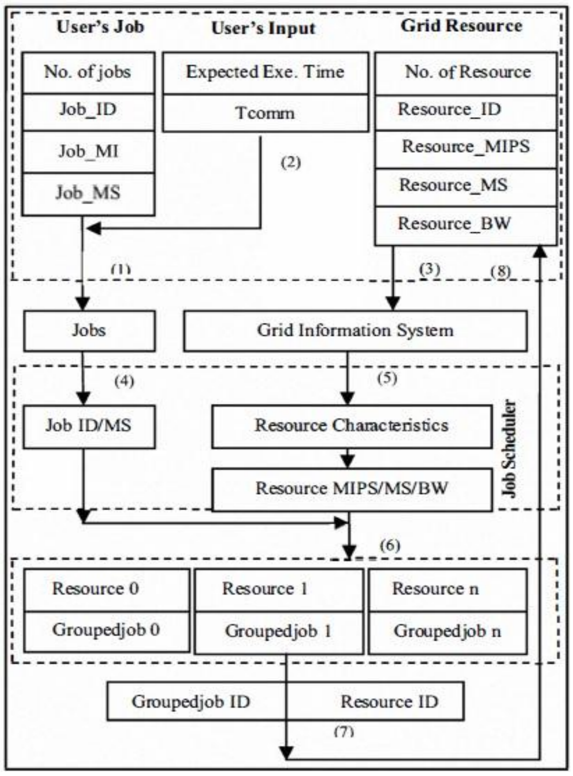

Figure 2. Architecture of memory-dependent job scheduler

B.Proposed Memory-Dependent Grouping Strategy

Grouping strategies is done based on the resources status; according to processing capabilities (in MIPS), bandwidth (in Mb/s), and memory size (in Mb) of the available resources.

Each job is described by four tuple constraints:

$$Job_j = (\tau_{ej}, \alpha_{cj}, \pi_{mj}, \tau_{tj}) \qquad (1)$$

Where τe denotes the expected execution time of the job j, α_c denotes the amount of computation in MI of the job j, π_m denote the amount of code and data memory needed during the execution of the job j, τ_t denote expected Transfer time of the job j to the selected resource and $\tau_{ej} >= \tau_{tj}$. Similarly,

Each resource is described by three tuple constraints:

$$R_i = (\rho_{pi}, \beta_{bi}, \pi_{mi}) \qquad (2)$$

Where ρ_p denotes the processing power of the resource i in MIPS, β_b denotes the network bandwidth of the resource i, π_m denote the amount of memory available at the resource i.

Each grouped job can be described by four tuple constraints:

Grouped-job = (T_{EG}, C, M, T_{TG}) (3)

Where T_{EG} is the total execution time of grouped jobs, C is total amount of computation of the grouped jobs, M is the total memory requirement of the grouped jobs and T_{TG} is the total transfer time of the grouped jobs.

Therefore,

$T_{EG}= \sum\tau_{ej}$,

$C=\sum\alpha_{cj}$,

$M=\sum\mu_{mj}$,

$T_{TG}=\sum\tau_{tj}$, and

$T_{EG} >=T_{TG}$.

To evaluate the make span, an analytical performance model is defined in terms of communication and computation cost of the grouped jobs.

$T_G = T_{commG} + T_{compG}$

$T_{commG}=t_E+\Sigma\tau_{ej}$,

$T_{compG}=t_S+\Sigma\tau_{tj}$,

Where T_{compG} is the total computation time of the grouped jobs, T_{commG} is the total communication time of the grouped jobs, Where t_E is computation startup time and t_S is network startup time.

The grouping strategy of this proposed model must satisfy the following conditions:

$\Sigma J=1$ to K $\alpha_{cj}<=\rho_{pi}*\Sigma J=1$ to K τ_{ej} (4)

$\Sigma J=1$ to K $\mu_{mj}<=\pi_{mj}$ (5)

$\Sigma J=1$ to K $\mu_{mj}<=\beta_{bj}*\Sigma J=1$ to K τ_{tj} and (6)

$T_{compG}/T_{commG}>=1$ (7)

Where, K is number jobs selected in a group.

Equation (4) specifies the processing requirement of the Grouped job or coarse-grained job shouldn't exceed to the resource processing capability at any point of time during grouping of the jobs. Equation (5) express a Memory size requirement of the groupedjob shouldn't exceed to the resource memory size capability. Equation (6) specifies a Memory size of the grouped jobs shouldn't exceed to resource transfer capability at any point of time during grouping of the jobs. Equation (7) specifies that communication time of the grouped jobs should not exceed computation time of the grouped jobs. These are the main constraints in job grouping strategy that influences the way job grouping is performed to achieve minimum job execution time and maximum resource utilization in the Grid system.

C. Memory-Dependent Job Scheduling Algorithm

```
Groupedjob_k=0, j=0, k=0;
Sort resources in descending order according to their MIPS;
Jobs are taken in FCFS order and assigned each an ID;
 For i:=0 to ResourceList_size-1 Do
 {
  $T_{EG}$=0,C= 0, M=0, $T_{TG}$=0;
  while ( j = Joblist_size-l)
  {
        $T_{EG} = +\tau_{ej}$ ;
        $C = +\alpha_{cj}$ ;
        $M = +\mu_{mj}$;
        $T_{TG} = +\tau_{tj}$;
        $Ri_{MI} = \rho_{pi} * T_{EG}$;
        $Ri_{BW} = \beta_{bi} * T_{TG}$;
 if((($C<=Ri_{MI}$) && ($M <=\pi_{mi}$)) && (($M<=Ri_{BW}$)))
 {
        $Groupedjob_k= Groupedjob_k+ Job_j$;
        j ++;
 }
 Else
{
        $T_{EG}=T_{EG}-\tau_{ej}$ ;
        $C=C-\alpha_{cj}$;
        $M=M-\mu_{mj}$;
        $T_{TG}=T_{TG}-\tau_{tj}$;
 }
 Submit the Groupedjob_k to R_i
 Set computational power of the resource R_i to zero
 k++;
 Break;
}
If (Prearranged Time of Reconstruction is arrived) then
{
 Reconstruct the Job queue and Resource queue;
 Go to step 1;
}
}
End;
```

V. EXPERIMENTAL EVALUATION

A. Experimental Result

GridSim [11] has been used to create the simulation of grid computing environment. The simulation is conducted in heterogeneous environment to verify the improvement of proposed model over other scheduling models. 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 and transfer time. (See Table I below the page). The processing time is taken into account to analyze the feasibility of the proposed scheduling algorithm.

TABLE I. COMPARISION BETWEEN ALGORITHMS ACCORDING TO THEIR PROCESSING TIME

NO.OF JOBS	PROCESSING TIME		
	MDJS	*AGJS*	*DJGBSDA*
100	62	81	92
200	144	168	195
300	183	234	254
400	257	335	476
500	274	396	529

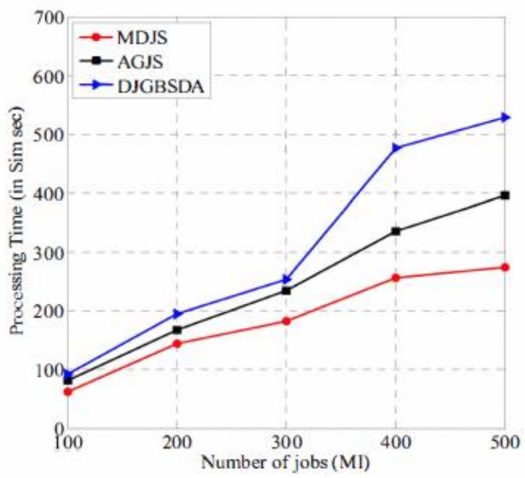

Figure 3. Processing time of the algorithms

B. Experimental setup and comparison

Simulation is conducted to analyze and compare the processing time of jobs between proposed MDJS algorithm with AGJS [9] and DJGBSDA [6] algorithms. Result shows that proposed scheduling model is significantly reducing the processing time as the number of job increases in comparison to others.

VI. CONCLUSION AND FUTURE WORK

This proposed "A Memory-Dependent Dynamic Job Scheduling Algorithm in Grid Computing" is a grouping based job scheduling strategy that has taken into account memory constraint of individual jobs together with expected execution and transfer time at the job level rather than at the group level.

The simulation results have shown that the proposed model is able to achieve the mentioned objectives in grid environment. The comparative study shows that the proposed AMDJS gives better performance than AGJS and DJGBSDA in terms of processing time. The proposed model provides a real time grid computing environment and also reduces the waiting time of the grouped jobs. The complexity of the of the proposed memory-aware dynamic job scheduling is O(nlogn) considering sorting of resources according to their MIPS and without which it will run in linear time.

In future, this work can be extended to design a high performance parallel scheduler for grid system to realize a real grid environment.

REFERENCES

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

[2] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, vol. 15, No. 3, 2001.

[3] Berman, F., Fox, G. and Hey, A.: Grid Computing Making the Global Infrastructure a Reality. London,Wiley, 2003.

[4] Foster, I. and Kesselman, C. Computation Grids. Foster, I. and Kesselman, C. eds. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999, 2-48.

[5] Ian Foster and Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Elsevier Inc., Singapore, Second Edition, 2004.

[6] 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, pp. 41-48, 2005.

[7] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan, Utpal Chandra Dey, "An Agent based Dynamic Resource scheduling Model with FCFS-Job Grouping Strategy in Grid Computing", WASET, ICCGCS 2010., in press.

[8] 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, pp. 117-126, 2006.

[9] 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, pp. 556-559, 2009.

[10] 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.

[11] R. Buyya and M. Murshed, GridSim; A toolkit for the modeling and simulation of distributed management and scheduling for grid computing, 2002.