World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:4, 2010

An Agent Based Dynamic Resource Scheduling Model with FCFS-Job Grouping Strategy in Grid Computing

Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan, Utpal Chandra Dey

***Abstract*—** Grid computing is a group of clusters connected over high-speed networks that involves coordinating and sharing computational power, data storage and network resources operating across dynamic and geographically dispersed locations. Resource management and job scheduling are critical tasks in grid computing. Resource selection becomes challenging due to heterogeneity and dynamic availability of resources. Job scheduling is a NP-complete problem and different heuristics may be used to reach an optimal or near optimal solution. This paper proposes a model for resource and job scheduling in dynamic grid environment. The main focus is to maximize the resource utilization and minimize processing time of jobs. Grid resource selection strategy is based on Max Heap Tree (MHT) that best suits for large scale application and root node of MHT is selected for job submission. Job grouping concept is used to maximize resource utilization for scheduling of jobs in grid computing. Proposed resource selection model and job grouping concept are used to enhance scalability, robustness, efficiency and load balancing ability of the grid.

***Keywords*—**Agent, Grid Computing, Job Grouping, Max Heap Tree (MHT), Resource Scheduling.

I. INTRODUCTION

THE word "grid" was first coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering projects [1]. The aim of grid computing is to aggregate the computing power of widely distributed resources, and provide non-trivial services to users [2]. Grid computing is a virtual computing having collection of clusters. Virtual computing refers to the system having features of choosing a resource in some specific manner and submits jobs on it. Whereas cluster means more than one node connected to each other either within a cabinet or over a LAN, giving users a single system image [3] and having various important facilities as scalability, high throughput, and high performance. Grid computing facilitates large scale resource sharing resulting in high speed job execution with less cost.

Authors are with School of Computer Engineering , Department, Kalinga Institute of Industrial Technology, University, Bhubaneswar, India
(e-mail: { rakshasharma10, vishnu.soni132, manojku.mishra05, prachetbhuyan}@gmail.com).

Both Resource and job Scheduling are two challenging tasks in grid computing, therefore efficient utilization of computational resources requires a well coordinated resource management system and an optimal job scheduling strategy. In addition for efficient resource utilization, it is important to consider processing time of jobs. To achieve the mentioned objectives, various works has been done by many researchers, still more research needs to be done to further improve the performance of scheduling algorithm in grid computing. As a result, this paper proposes a new approach, which is "An Agent Based Dynamic Resource Scheduling Model with FCFS-Job Grouping Strategy".

This proposed paper consists of two major parts, first part gives an agent based dynamic resource scheduling model that manages the grid resources by considering the real time available computational ability of each cluster at a particular instance and selects the cluster with highest computational power 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. The objective of this paper is to design a model for scheduling grid resources which minimizes the processing time for resource selection and an efficient job scheduling strategy to reduce processing time of jobs.

The rest part of this paper is organized as follows. In section II, related works of various research papers and their analysis in this field are discussed. Section III presents the proposed model for resource scheduling and FCFS-job grouping strategy. Section IV provides simulation and experiments of the proposed model with help of GridSim toolkit and results shows the improvement in processing time of jobs in comparison to other model. Finally, Section V gives conclusion and future work and lastly references.

II. RELATED WORK

Resource management and job scheduling are two most important and difficult issues in grid computing environment. In this field several works has been done. Research on novel dynamic agent based resource management and job scheduling algorithm in which highest computational power resource is selected through Heap Sort Tree for scheduling of the jobs, but this resource scheduling strategy does not provide a real time grid environment [4]. An Agent based Resource Management that gives an alternate solution at the situation when resource discovery fails. This algorithm identifies an equivalent resource through negotiation in case resource discovery fails without affecting the performance and it also avoids unnecessary resource discovery, but drawback of this paper is that time taken to find the resource with negotiation solution is more as a result waiting time of the job increases [5], [6]. A job scheduling algorithm for maximum processor utilization, throughput and minimum turnaround time to achieve performance optimization. It uses backup node in the case of supervisor node fails, but it does not utilize resource sufficiently and the processing time of the job is high [7]. Grouping-based job scheduling algorithm that groups the jobs according to MIPS of the resource. This model reduces the processing time of the job and communication time, but the algorithm doesn't take the dynamic resource characteristics into account and the grouping strategy can't utilize resource sufficiently [8]. Grouping-based job scheduling algorithm that groups the jobs according to MIPS and Bandwidth of the resource. These scheduling strategies reduces the processing time of the jobs and network delay but do not utilize resource sufficiently and preprocessing time to group the jobs is more [9], [10]. Hierarchical job scheduling approach using two level scheduling consisting of top level (global scheduling) and local level. It tries to reduce overall turnaround time and maximize system utilization for high system loads. Main drawback of this paper is, it does not consider the dynamic behavior of the grid resources and the algorithm runs on homogenous environment [11].

To overcome some of the above problems the proposed model "An Agent Based Dynamic Resource Scheduling with FCFS-Job Grouping strategy" is presented. This tries to provide real time computing ability of the selected resource by submitting the grouped job as soon as group is formed and this also reduces the waiting time of the grouped jobs. The resource selection uses MHT algorithm to select the resource which takes O(logn) time. This paper also reduces the processing time of the jobs and utilizes the resource more sufficiently.

III. Resource Scheduling Model

A. Basic Grid system

Grid computing is based on access to remote computing resources, which may be a group of clusters connected over high-speed networks. Cluster is the collection of two or more nodes connected together over a LAN and one node is selected to supervise all other nodes within the cluster giving it a single system image at the grid system. Proposed model is based on agents. Agent can improve the efficiency of resource management and scheduling system by reducing time of resource selection. They can be able to communicate with each other so as to give real time dynamic services for scheduling application.

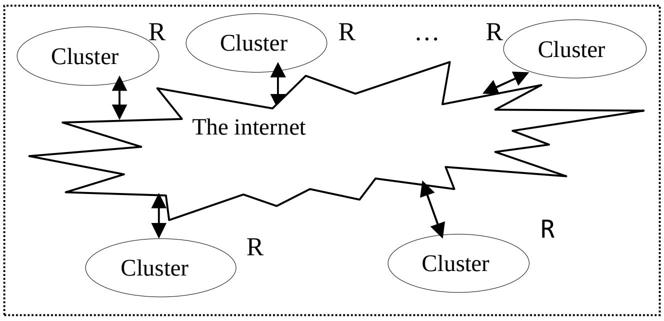

Fig. 1 Clusters connected across network
(R denotes combined resources in MIPS available in a cluster)

B. An Agent Based Dynamic Resource Scheduling Model

The execution of a job in a dynamic environment like Grid often calls for efficient algorithms to schedule the resources required for successful execution of the jobs. These resources may themselves be dynamic and may enter or leave the system at any point of time or fail and can be idle. For these reasons resource scheduling in large-scale grids can be very challenging. So an efficient resource scheduling strategy is required to obtain real time computational ability of the grid resources and minimize total execution time of jobs. So a scheduling strategy is required to generate schedules, which seek to minimize the total execution time of jobs and also adapt to the heterogeneity and the dynamism of the environment.

Proposed agent based dynamic resource scheduling model is a hierarchical two-layered approach, the top-layer is called grid level (global level) and other layer is called cluster level (local level). In this proposed model an agent works in all the nodes of each cluster in the grid system. The agent has two different responsibilities and functionalities as: the agent running in each node of every cluster in the grid system is called Executer agent and respective nodes are called as executer nodes. Responsibility of an executer agent is to obtain computational power of the node in which it is running. The nodes within each cluster are sorted according to their computational power. The node with highest computational power in each cluster is selected as the supervisor node of the respective clusters. The executer agent running in respective supervisor nodes of each cluster is called supervisor agent. This supervisor agent running in supervisor node of a cluster gives it a single system image at the grid level and represents the clusters during resource scheduling in the grid system. Supervisor agent of a selected cluster receives grouped-jobs from job scheduler and distributes it to all other executer nodes according to their capability under its supervision.

In order to achieve good load balance, supervisor agent of the cluster distributes the tasks of the received jobs according to computational ability of individual executer nodes within the same cluster. The executer node with second highest computational power in each cluster acts as a backup node incase the supervisor node fails. This backup node performs the works of a normal Executer node when the supervisor node is active. Therefore, this arrangement makes the proposed grid model fault tolerant. Executer agents running in nodes of a cluster works in local level or cluster level and the executer agent running in supervisor node known as supervisor agent works in global level or grid level.

Functions of an Executer Agent:

(1) Whenever prearranged time arrives these agents are activated in all available nodes of clusters within the grid system.
(2) Dynamically obtains required characteristics of the resources available in every node within each cluster of the grid system.
(3) The node with highest computational power is selected as the supervisor node by sorting the nodes according to their MIPS and the executer agent running in that supervisor node is called supervisor agent.
(4) Receives the tasks from the supervisor agent and performs the execution of the assigned tasks.
(5) Sends the results of the assigned tasks back to the supervisor agent.
(6) The executer agent running in the node having second highest computational power acts as backup and assumes the role of a supervisor in case the first supervisor node fails.

Functions of a Supervisor Agent:

(1) Controls and coordinates the activities of all other executer nodes of the cluster in the grid system.
(2) Obtains total computational power in MIPS of a cluster by combining its own and the individual computational power of executer nodes and gives the cluster a single system image at the grid level.
(3) Competes in the selection of highest computational power cluster by using Max Heap Tree at the grid level.
(4) Receives a group of jobs from the job scheduler and different tasks of grouped jobs are scheduled locally among the executer nodes within that cluster for execution.
(5) Receives the result of completed tasks from executer nodes and accumulates the results of corresponding jobs in the job group.
(6) After completion of the jobs results are sent to the requesters.

Supervisor node selection at Local level:

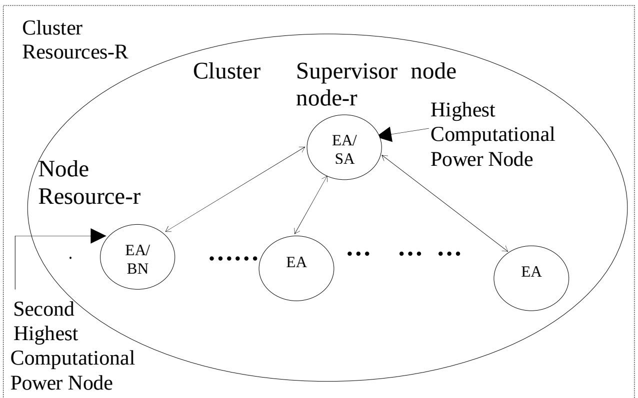

Fig. 2 Supervisor node selection

Resource selection in Global Level:

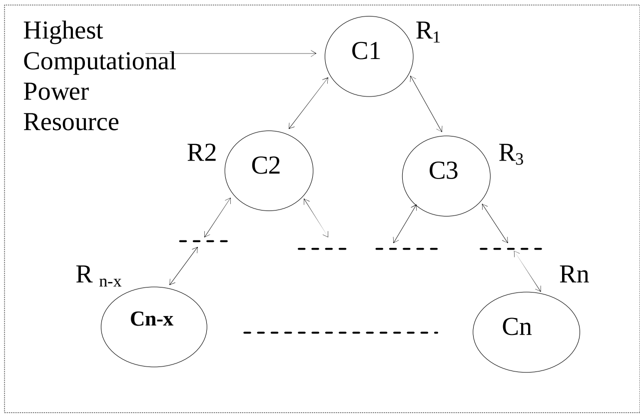

Fig. 3 Highest computational power resource selection

C. *Proposed FCFS-Job Scheduling Strategy*

Job scheduling is the mapping of jobs to specific physical resources, trying to minimize some cost function specified by the user. So an optimal scheduling strategy is required to generate schedules, which seek to minimize the total execution time of jobs and also adapt to the heterogeneity and the dynamism of the grid environment.

For job grouping, computational power of resource is taken as key constraint. Jobs are grouped in such a way to maximize the resource utilization. Scheduler selects a resource by MHT and multiplies the resource's Million Instructions per Second (MIPS) with the granularity size, which is the time within a job is processed at the resource. The value of this calculation produces the total Million Instructions (MI) for that particular resource to process within a particular granularity size [8]. The scheduler group's user jobs by accumulating the MI of each job matching with selected resource MI. If the MI of grouped jobs is more than the resource MI, the last MI added will be removed from the group job MI and finally grouped job is submitted to selected resource which will reduce communication time and total processing time.

If job submission is successful, result is sent to the user, if job submission fails for instance when resource fails, network connection down, scheduler is not ready or scheduler is ready but not accepting new submissions of jobs, bandwidth problem or resource failure condition etc, either job is resubmitted to the same resource or job is submitted to another selected resource by running the proposed dynamic resource selection algorithm. In case of repeated failure job is resubmitted to the same resource for a constant number of times after which job is submitted to a new selected resource either by removing or adding certain number of jobs according to the MI of resource.

The proposed algorithm is as follows:

1. A user sends job to the Grid Portal;
2. The Grid Portal sends a job to the Grid System;
3. Jobs come in first come first serve order;
4. Set K = 0 , R = 0;

5. If (Resource is not ready) then call MHT;
6. Select root node for Job Submission;
7. For (i=0 to Job_List_Size-1)
8. {
9. $R_{MI} = R_{MIPS}$ * Granularity size;
10. set $(Total\_Job)_{MI} = 0$;
11. While ($R_{MI} \geq (Total\_Job)_{MI}$)
12. {
13. Total_Job = Total_Job + Job_i ;
14. i++;
15. }
16. i--;
17. Total_Job = Total_Job - Job_i ;
18. Put Total_Job into $(JRQ)_K$ and submit Job from $(JRQ)_K$ to the selected Resource;
19. If (Job submission == success)
20. {
21. K++;
22. Set the computational power of selected Resource = 0;
23. }
24. Else
25. {
26. Call MHT for next Highest computational power Resource;
27. R++;
28. while ($R_{MI} \geq ((JRQ)_K)_{MI}$)
29. {
30. Total_Job = Total_Job + Job_i ;
31. i++;
32. $(JRQ)_K$ = (Total_Job);
33. }
34. i--;
35. while ($((JRQ)_K)_{MI} > R_{MI}$)
36. {
37. Total_Job = Total_Job – Job_i ;
38. i-- ;
39. $((JRQ)_K)$= Total_Job ;
40. }
41. i++;
42. Submit $(JRQ)_K$ into Resource and go to step 19 ;
43. } // End of Else;
44. Tell Portal Job is submitted successfully;
45. If (Prearranged Time of Reconstruction is not yet arrived) then Call MHT for next highest computational power resource;
46. } // End of for loop;

D. Example of the Proposed Job Scheduling Strategy

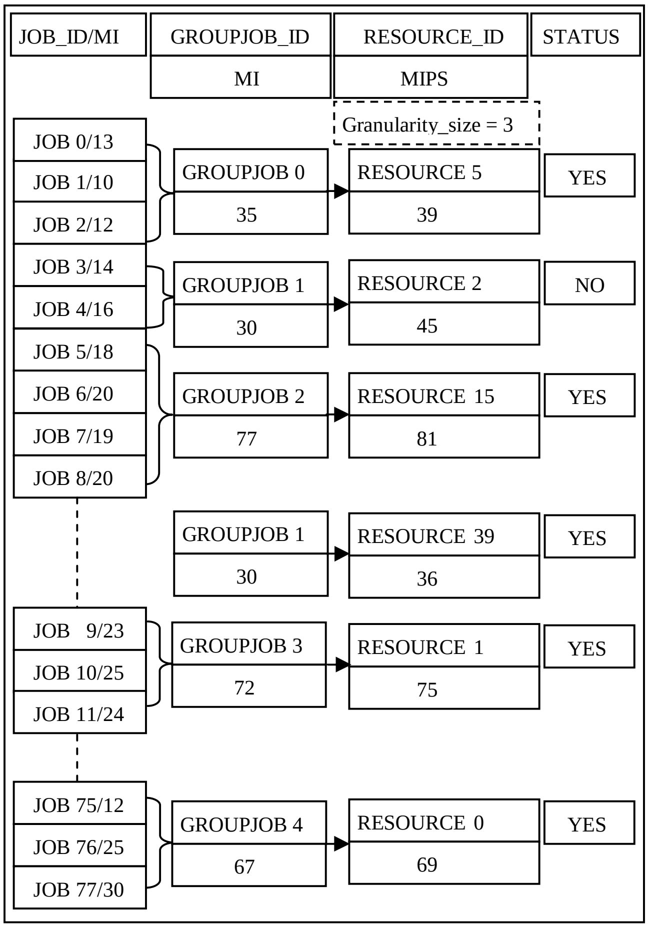

Fig. 4 Example of the proposed job scheduling strategy

E. Description of Proposed Model

1) Grid users submit jobs to grid system.
2) All available executer agents are activated in all nodes across the grid system to select supervisor node of each cluster at the local level. Supervisor node of each cluster gives highest computational power of that cluster.
3) Cluster with highest computational power is selected using MHT algorithm at grid level, which is the root node of MHT. (Clusters which have finished selecting their supervisor node are allowed to participate in constructing MHT)
4) Jobs from job-queue are taken in FCFS order and are grouped according to the computational ability of root node (i.e. cluster) of MHT.
5) Grouped-jobs are submitted to the selected root node of MHT, if job submission fails then resubmit the grouped-jobs to the same selected cluster. If job submission fails repeatedly for a constant number of times, call step 2 to achieve a new highest computational power cluster to resubmit that grouped-jobs.
6) Set the computational power of selected cluster to zero after successful submission of grouped-jobs to disappear its computational power from the rest part of grid.

7) Call step 2 periodically.

Normally, Resource selection is done periodically according to this proposed model but it also calls for resource selection in some special cases like when grouped-jobs is submitted to the resource, any fault occurs, new resource join grid, reconstructions time arrives.

IV. Experimental Evolution

A. Experimental setup and comparison

GridSim [12] has been used to create the simulation environment. A simulation is conducted in heterogeneous environment where each cluster has machines with different characteristics and MIPS, to verify the improvement of proposed model over other scheduling model. In this simulation, size of the granularity is taken as 3 seconds for both scheduling algorithm. The MIPS is the main factor to constrain the fine-grained to coarse-grained jobs. The processing time is taken into account to analyze the feasibility of our algorithm.

TABLE I
COMPARISION BETWEEN ADRBS AND DJGBS

Number of jobs	Processing Time	
	ADRBS	DJGBS
100	55	92
200	133	195
300	175	254
400	270	476
500	342	529

Fig. 5 Job processing time

B. Experimental Results

Simulation is conducted to analyze and compare the processing time of jobs between proposed algorithm ADRBS and DJGBS algorithm [8]. Result shows significant reduction in processing time of ADRBS algorithm when number of jobs increases in comparison with DJGBS algorithm.

V. Conclusion and Future Work

In order to reduce processing time and utilize grid resource sufficiently, An Agent Based Dynamic Resource Scheduling with FCFS-Job Grouping strategy model has been proposed. The simulation environment has shown that the proposed model is able to perform job and resource scheduling in grid environment. The proposed model has also shown comparative result in better scheduling compared to DJGBS model. The proposed model provide real grid computing environment and reduces the time complexity.

In the future, we are planning to implement job and resource scheduling with genetic algorithm to increase the performance.

References

[1] Foster, and C. Kesselman, Globus: a metacomputing infrastructure toolkit, International Journal of High Performance Computing Applications, Vol. 2, pp. 115–128, 1997.

[2] 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.

[3] Rajkumar, Buyya: Architecture Alternatives for Single System ImageClusters, Conference on High Performance Computing on Hewlett- Packard Systems (HiPer'99), Tromse, Norway, 1999.

[4] Fufang Li, Deyu Qi, Limin Zhang, Xianguange Zhang and Zhilli Zhang, "Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing", Proceedings of the IEEE first International Multi-Symposiums on Computer and Computational Science, IEEE, 2006.

[5] Ms.P.Muthuchelvi, Dr.V.Ramachandran, "ABRMAS: Agent Based Resource Management with Alternate Solution", The Sixth International Conference on Grid and Cooperative Computing IEEE, 2007.

[6] Junyan Wang, Yuebin Xu, Guanfeng Liu, Zhenkuan Pan, and Yongsheng Hao, "New Resource Discovery Mechanism with Negotiate Solution Based on Agent in Grid Environments", The 3rd International Conference on Grid and Pervasive Computing, Workshops, IEEE, 2008.

[7] Homer Wu,Chong-Yen Lee,Wuu-Yee chen,Tsang Lee, "A Job schedule Model Based on Grid Environment", Proceeding of the First International Conference on Complex, Intelligent and Software Intensive System, IEEE, 2007.

[8] Nithiapidary Muthuvelu, Junyang Liu, "A Dynamic Job Grouping-Based Scheduling for Deploying Application with Fine-Grained tasks on Global Grids", Vol. 44, Australasian Workshop on Grid Computing and e-Research, AusGrid -2005.

[9] Quan Liu, Yeqing Liao, "Grouping based Fine-Grained job Scheduling in Grid Computing", First International Workshop on Education Technology and Computer Science, Vol.1,pp. 556-559, IEEE, 2009.

[10] Ng Wai Keat, Ang Tan Fong, Ling Teck Chaw, Liew Chee Sun, "Scheduling Framework For Bandwidth-Aware Job Grouping-Based Scheduling In Grid Computing", Vol.19(2), pp. 117-126, Malaysian Journal of Computer Science, 2006.

[11] J. Santoso, G.D. van Albada, B.A.A. Nazief, P.M.A. Sloot, "Hierarchical Job Scheduling for Clusters of Workstations", pp. 99-105. ASCI, June 2000.

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