A Comparative Analysis of Resource Scheduling Techniques in Grid Environment

Ramandeep Singh
Lovely Professional University
Phagwara, Punjab

Jyoti
Lovely Professional University
Phagwara, Punjab

ABSTRACT
Grid computing is an emerging technology that involves coordinating and sharing of resources to carry out complex computational problems. Resource Allocation is a very challenging task in Grid Environment. Because there are ample amount of jobs and quick responses to the users are necessary in a real Grid Environment. A lot of techniques have been proposed for managing the allocation of resources. This paper represents comparative study of some of these techniques based on some parameters.

General Terms
Grid Computing, Resource Scheduling.

Keywords
Grid computing, Scheduler, Resource Allocation, Resource Utilization Rate, Load Balancing.

1. INTRODUCTION
With the increase in the usage of internet/web applications and availability of high powerful computers, the way we do computing is changing rapidly. The computations of tasks are not only limited to individual computers. The organizations are showing interest in geographically dispersed machines for solving their complex applications which are beyond the processing capabilities of individual computers [1]. Grid computing is emerging as a new paradigm for solving complex scientific and engineering problems. Basically, it is a form of distributed computing that enables the users to share the widely distributed, heterogeneous resources connected through the network to carry out their complex computational tasks. It is analogous to the electrical grids. As in electrical grids, the users get access to electricity without knowing where the power plant is or how the electricity is generated, in the same way users get power (resources) from the grid without having to bother about the source of the computing power.

Resources are generally scarce, so should be used optimally. There are a wide range of resources available in the grid including CPUs, storage elements, communication links, software, equipments etc that can be shared among different users for executing their jobs. In recent years, the researchers have proposed several efficient scheduling algorithms that are used in grid computing to allocate grid resources, categorized into centralized and decentralized approaches. This paper is organized as Section 2 describes the grid model. Section 3 describes the various resource allocation techniques and Section 4 represents the analysis and comparison of different techniques on the basis of different parameters and section 5 describes the conclusion, lastly Section 6 describes references.

2. GRID MODEL
Grid computing is a combination of clusters which contains nodes, also called hosts which provide their computational resources for execution of jobs. The grid architecture consists of four main parts: Scheduler, Users, Grid Information System (GIS) and resources. Scheduler is responsible for locating a machine on which to run a grid job that has been submitted by a user. GIS keeps track of the information of available resources in the grid, their current utilization rates, etc. It also provides resources registration service [2]. Firstly the local scheduler finds the required resources locally in the cluster. If resources are not available in local cluster, then local scheduler sends a request to neighbour schedulers and request for the required resources. The remote resources are then under the control of the local scheduler [3] as shown in Figure 1.

3. RESOURCES SCHEDULING TECHNIQUES
Grids are complex structures. Scheduling of resources is difficult due to the reasons that they are widely distributed and heterogeneous in nature. Grid environment is highly dynamic in nature. There are great chances that the resource availability changes during the job execution. This may include failure of allocated resources, communication links, schedulers etc. A good Grid resource scheduling method should be distributable, extendable and should have fault-tolerant mechanism [4].

From recent years, a lot of research has been done for efficient utilization of resources. Resources allocation techniques are of two types: static and dynamic. In static allocation, the jobs are assigned to the processors/ resources before their execution begin. The basic motive of this approach is to minimize the turnaround time of jobs, thus decreasing the communication and execution time. Whereas in dynamic scheduling, jobs are assigned immediately prior to its execution, which may be after the jobs have commenced. The motive of this approach is to balance the load among the nodes.

In [5], the scheduling techniques are broadly classified into centralized and decentralized approaches on the basis of the duties of the Grid Scheduler. Centralized Approach is static in nature so not flexible to meet the changing requirements of the jobs during execution. Static scheduler strategy is used for distribution of resources among jobs. This approach cannot guarantee that jobs complete within the deadlines when the jobs increase in grid environment. This approach includes DAG node weight based policies which allocate the resources according to their pre-assigned weights [6,7] and the cluster based policies which allocate same resource to tightly coupled tasks in order to reduce communication overhead [8]. Represented by flat tree, not a suitable structure because master node become overloaded very easily whereas in decentralized Approach, several nodes are in control and can be represented by a non-flat tree. Each node can control a subset of processors in order to prevent bottlenecks. Dynamic in nature and allows to accommodate the changing requirements of jobs during their execution times. This approach combines the job scheduling and job assignment process into single process. Software agents such as learning agents are widely used in dynamic grid environment [9,10]. But the deficiency in this is that synchronization issues associated with jobs have not been considered.

[11] discussed Constraint-based Job and Resource Scheduling (CBJRS) approach which uses jobs grouping strategy to allocate resources. Resources are arranged in hierarchical manner using Heap Sort, with the highest computational power resource at the top. Jobs are put into groups until they meet following constraints.

1.Groupedjob_MI≤ Resource_MIPS*Granularity size

2. Groupedjob_MS≤ Resource_MS

3. Groupedjob_MS ≤ Resource_baud_rate*T_{comm}

The topmost resource is fetch and the jobs are assigned to it until they meet the specified constraints. When the conditions violate, the next resource is fetch and next jobs are then assigned to it as per constraints and this process continues till all the jobs get required resources. The proposed algorithm increases the resources utilization rate as a single resource can be assigned to multiple jobs, improves processing time and cost as compared to AFJS, DJGBSDA. This algorithm is non-preemptive and dynamic in nature. The deficiencies in this model are that it is not suitable for executing jobs in parallel thus decreasing throughput, does not tell whether the job submission fails and does not consider jobs dependencies while scheduling them.

[2] proposed an Improved Resource Scheduling Approach using Job Grouping Strategy which uses a hierarchical approach for resource and job scheduling. Global scheduler searches the cluster with highest computational power. The selected cluster is then used to schedule the job. The next available job is then scheduled to the same cluster if the requirements of first and second job are within the capabilities of the selected cluster. If it is not so, the next highest power cluster is selected for scheduling. Local scheduler is responsible for searching computational power of all nodes within the cluster. Some jobs are scheduled according to best fit policy on a particular host and rest according to round robin fashion unless they are blocked. Thus, Global scheduling makes load balanced across clusters and local scheduling within the cluster. Jobs are scheduled in FCFS manner and this approach is non-preemptive in nature at global level and preemptive at local level. Thus, higher priority jobs need to wait until the resources are released by lower priority jobs. Improves processing time and cost of resources allocation to different jobs when number of jobs in the environment increases.

[12] discussed on a hybrid resource allocation technique in grid environment. This approach presents resource allocation as a special case of linear programming transportation problems. Grid resource allocation deals with the assignment of tasks from number of jobs to a number of processors at minimum transportation cost. This approach assigns resources to different jobs on the basis of the allocation cost. Each job is divided into equal sized tasks. These tasks are distributed among different processors based upon their capabilities and tasks' requirements. The main objective of this is to minimize the allocation cost with certain set of constraints. These constraints ensure that the total processing units allocated to processors should not exceed their capabilities and all tasks of a given job should get resources. This approach is based on DLT (Division Load theory). This paper also showed that this hybrid approach is better than other techniques such as DLT, NWCM, LCM etc. Hybrid allocation favors the equal distribution of jobs. But the deficiencies in this algorithm are that resources are not utilized efficiently and it has no fault tolerance mechanism to detect whether job submission fails or any error occurs during execution. It is not suitable for executing those jobs which cannot be divided into sub jobs.

[13] discussed an ant algorithm of resource scheduling among different jobs. It uses a heuristic approach which tries to find out the best possible solution. It allows allocating resources dynamically and is flexible enough to accommodate changing requirements of the jobs. This algorithm includes a resource state prediction mechanism for proper task and resource scheduling. Results showed in paper indicated that this algorithm performs well with respect to response time and utilization rates. This algorithm guarantee good load balancing of the machines .The advantages of it are that it allows jobs to execute in parallel and have scalability features. But the deficiency is that it is not tested in real time Grid Environment.

4. ANALYSIS AND COMPARISON

This section represents the comparison of various resource allocation techniques based on certain parameters. CBRJS is an efficient algorithm in term of processing cost and resource utilization. Similarly IRSJGS provides high resource utilization rate and a fair execution time. But both of these algorithm are not robust and scalable as they do not provide any mechanism to check whether the job is submitted successfully or results have been computed correctly. An Ant Algorithm provides parallel execution of jobs but resources are underutilized and also do not have fault tolerance mechanism. Hybrid scheduling algorithm is a good resource scheduling algorithm in terms of processing cost, resource utilization, fault tolerance etc. Thus this approach is dynamic, robust, scalable, and distributable with average load balancing. Comparison of these techniques is illustrated in Table 1.

5. CONCLUSION

This paper provides a comparative study of some of the existing techniques of resource scheduling among different jobs. Resource Scheduling is a very difficult task due to heterogeneity of resources. Static techniques bound resources to the jobs before execution. Thus resources are not utilized efficiently and not able to meet the changing requirements of the jobs. Dynamic techniques helps in load balancing as jobs can be migrate from one resource to another in case of bottleneck. Dynamic techniques are more suitable in grid environment as compared to static approaches.

Load balancing and Resource Allocation are interrelated. Even advance resource allocation techniques may suffer due to improper load distribution among the available resources in Grid environment.

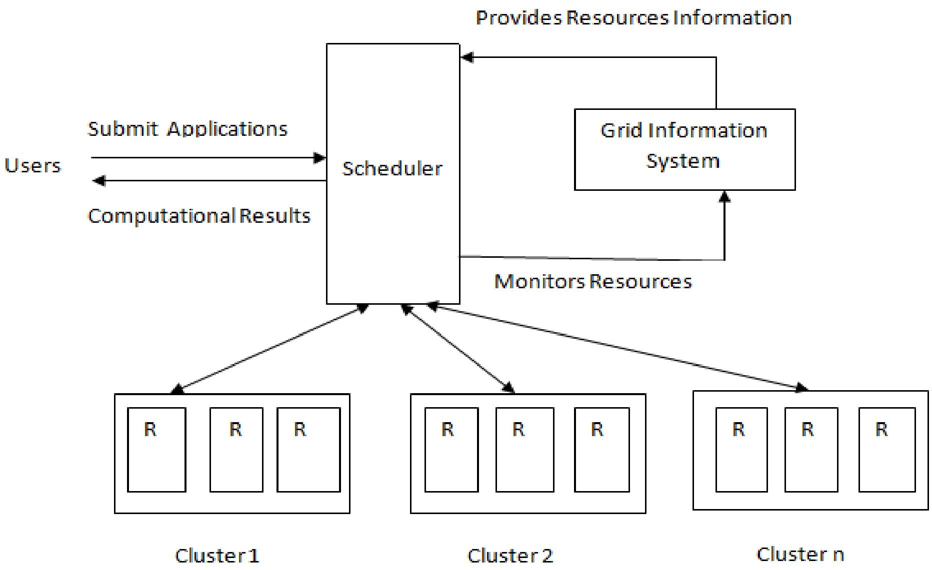

Fig 1: Basic Grid Model

Table1. Comparison of Techniques

Approach/ Parameters	CBRJS	IRSJGS	Hybrid Scheduling	Ant Algorithm
Processing time	Low	Avg	Avg	Avg
Mode	NP	P at local level	NP	P
Processing Cost	Low	Avg	Low	Avg
Execution	Serially	Serially	Parallel	Parallel
Fault tolerance	-NA-	-NA-	Low	-NA-
Resource Utilization	High	High	High	Low
Load Balance	Low	High	Avg	-NA-

Abbreviations: P- Preemptive, NP- Non-Preemptive

6. REFERENCES

[1] I.Foster and C.Kesselman. 2004 The Grid: Blueprint for a New Computing Infrastructure.

[2] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra 2010 An Improved Resource Scheduling Approach Using Job Grouping Strategy in Grid Computing

[3] Leila Ismail 2007 Dynamic Resource Allocation Mechanisms for grid computing Environment

[4] Quinn Snell, Kevin Tew, Joseph Ekstron, Mark Clement 2002. An Enterprise-Based Grid Resource Management System.Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11.

[5] J. Palmer, I. Mitrani. 2004. Optimal Tree Structures for Large-Scale Grids. Proceedings of the UK e-Science All Hands Meetings, Nottingham.

[6] R. Bajaj, D.P. Agrawal 2006. Improving scheduling of tasks in a heterogeneous environment. IEEE Transactions on Parallel and Distributed Systems.

[7] Z. Shi, J.J. Dongarra 2006. Scheduling workflow applications on processors with different capabilities. Future Generation Computer Systems

[8] S.N.Mehmood Shah, Ahmad Kamil Bin Ahmood and Alan Oxley. 2010. Modified Least Cost Method for Grid Resource Allocation.

[9] A. Galstyan, K.. Czajlowski, et al. 2005. Resource allocation in the grid with learning agents. Journal of Grid Computing.

[10] C. Zhu, Z. Liu, et al. 2004. Decentralized grid resource discovery based on resource information community. Journal of grid Computing

[11] Vishnu Kant Soni, Raksha Sharma, Manoj Kumar Mishra, Sarita Das. 2010. Constraint Based Job and Resource Scheduling in Grid Computing.

[12] Syed Nasir Mehmood Shah, Ahmad Kamil bin Mahmood and Alan Oxley.2010. Hybrid Resource Allocation Method for Grid Computing. Second International Conference on Computer Research and Development.

[13] Zhihong Xu, Xiangdan Hou, Jizhou Sun. 2003. Ant algorithm-based Task Scheduling in Grid Computing.