IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. IV (Mar – Apr. 2015), PP 13-16
www.iosrjournals.org

Comparative Analysis of Various Grid Based Scheduling Algorithms

Ms. Maniza Hijab[1], Dr. Avula Damodaram[2], Dr. Uma N. Dulhare[3],
Ms. M. Tahseen[4,]

[1,3,4]Information Technology Department, Muffakham Jah College of Engg.& Tech, Banjara Hills, Hyderabad -155,A.P. India.
[2]CSE Department, JNTUH & Director Academic Audit Cell, Jawaharlal Nehru Technological University Kukatpally, Hyderabad-72, A.P. India.

Abstract : *Grid computing provides access to heterogeneous resources which are distributed geographically which makes resource scheduling a complex problem. Hence, scheduling algorithms are necessary which allocate jobs to resources by taking into account the requirements of grid environment. The aim of scheduling is to achieve highest possible throughput by matching the need of the application with the computing resources available. In this paper, we review the various resource scheduling algorithms and discuss their pros and con.*

Keywords: *Grid Computing, Grid Scheduling, Scheduling Parameters, Scheduling Algorithms*

I. Introduction

The Grid consists of various components and resources. Grid Computing is an important paradigm that supports sharing and access to a large amount of computational and storage resources which are heterogeneous and geographically distributed. Grid computing has the design goal of solving problems too big for any single supercomputer, while retaining the flexibility to work on multiple smaller problems. Therefore Grid computing provides a multi-user environment. Its secondary aims are better exploitation of available computing power and catering for the intermittent demands of large computational exercises [1].The basic grid model is generally composed of a number of hosts, each composed of several computational resources, which may be homogeneous or heterogeneous. The four basic building blocks of grid model are user, resource broker, grid information service (GIS) and resources.

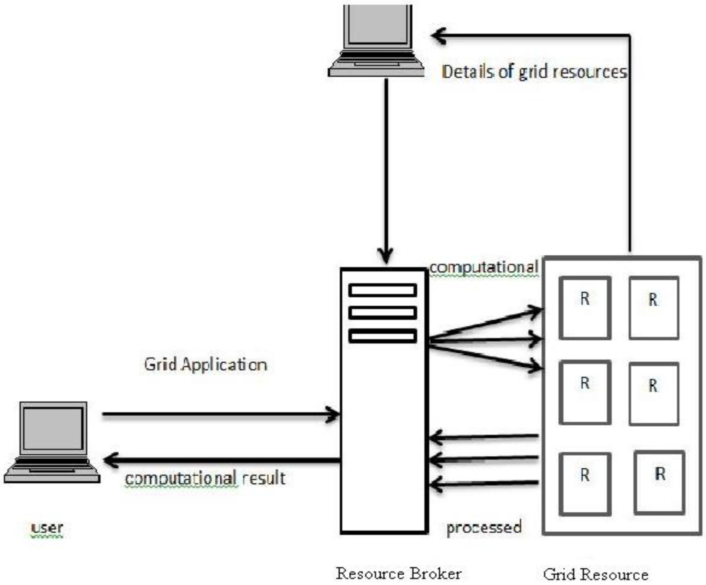

Fig.1 Basic Resource Model [4]

When user requires high speed execution, the job is submitted to the broker in grid. Broker splits the job into various tasks and distributes to several resources according to user's requirements and availability of resources. GIS keeps the status information of all resources which helps the broker for scheduling [2]. Appropriate and efficient allocation of tasks to the resources is called as job scheduling. Main aim of job scheduling is to provide proper load balancing of nodes and reduce the response time by proper allocation of jobs.

1.1. Types of Grid Scheduling[3]

A. Static versus Dynamic

In static scheduling, resource information and task information is available before scheduling. In dynamic scheduling, task allocation is done during its execution.

B. Centralized versus Decentralized

In centralized scheduling, a single scheduler is responsible for task scheduling whereas in decentralized scheduling multiple distributed schedulers are responsible for scheduling.

C. Hierarchical

In hierarchical scheduling two schedulers are used, one at global level and the other at local level.

1.2. Comparison parameters

Various parameters discussed in this paper are:

Completion Time: Time taken by a machine in grid to complete a task.
Sufferage Value: The difference between second earliest completion time and the earliest completion time.
Task Priority: The tasks with highest priority are given more weightage.
Computation Costs of Task: cost to execute a task by a processor.
Communication Cost: cost of communication between two tasks.
Turnaround Time (TT): It is the total time taken between the submission of a task for execution and the return of the result to the user.
Response Time (RT): It is the total time it taken to respond to a request. The response time is the sum of time taken by the task waiting in queue and the service time.

II. Existing Scheduling Algorithms

2.1Min-Min Algorithm

In [5] the author proposed an algorithm which finds the machines on which a task is completed in minimum time. This results in task-machine pairs with minimum completion time. Among all the task-machine pairs the pair with minimum completion time is selected. Once the task is assigned to the machine it is removed from the task list and the ready time of the machine is updated. This process is repeated until all the tasks are completed.

Advantage: The response time is reduced as the minimum completion time of job is considered.

Disadvantage: Estimating the minimum completion time is difficult and may not be accurate. Time consuming process.

2.2 Sufferage algorithm

In [6] the author proposed an algorithm which finds the machine on which a task is completed in minimum time. The Sufferage value is calculated which is the difference between the second earliest completion time and the earliest completion time. If the machine is unassigned then assign the task to the machine and remove the task from task list. Otherwise, if the Sufferage value of a task already assigned to the machine is less than the Sufferage value of the task selected then unassign the assigned task and add it to task list, assign the selected task and remove it from task list. The ready times of all machines are updated.

Disadvantage: Determining the earliest completion time and Sufferage value is tedious and time-consuming process and may not be accurate

2.3 Heterogeneous Earliest Finish Time (HEFT) Algorithm

In [8] the author proposed an algorithm which makes use of a directed acyclic graph. The average computation costs of tasks and the communication costs of edges are calculated.The rank of all the tasks is computed by traversing the graph upward. The tasks are sorted in decreasing order of rank. For each processor the earliest finish time (EFT) of tasks is computed. A task is assigned to the processor which minimizes its EFT

Disadvantage: More time spent on calculating computation costs and communication costs.

2.4 Critical Path-On- a Processor (CPOP) Algorithm

In [9] the author proposed an algorithm which makes use of directed acyclic graph. The mean of computation costs of tasks and communication costs of edges are calculated. For each task two ranks are computed by traversing the graph both upward and downward. The priority of each task is assigned by adding the two ranks. This algorithm uses critical path of graph, obtained by adding the computation cost of tasks on the path with the inter-task communication cost.The entry task is selected as critical path task. The task immediately following the entry if having highest priority is selected as critical path task.A priority queue is maintained for the tasks. If the selected task is on critical path it is assigned to critical path processor which minimizes the cumulative computation cost of the task on the critical path. If no, then it is assigned to a processor which minimizes the EFT of the task and the priority queue is updated.

Disadvantages:

a) It is not suitable for more number of job allocations because finding priority of job is tedious one.
b) Higher turnaround time.
c) CPU and memory wastage.

2.5 Reliability Aware Scheduling Algorithm with Duplication of HDC System (RASD) Algorithm

In [7]the author proposed an algorithm which makes use of a directed acyclic graph. The average computation costs of tasks and the communication costs of edges are calculated. The rank of all the tasks is computed by traversing the graph from the exit task.The tasks are sorted in decreasing order of rank. For a task assigned to a processor find the reliability of communication edge between the task and its predecessor. For each processor repeat the process and compute EFT of a task on a processor and also the reliability of the task on the processor an the system overhead.

2.6 Hierarchical Job Scheduling for Clusters of Workstations (HJS)

This[2] hierarchical scheduling mode uses two levels of scheduling - top level(global scheduling) and local level(local scheduling). The global scheduler uses single or multiple queues for scheduling jobs using FCFS, SJF policy. The local scheduler uses only one queue for all types of jobs with any one policy FCFS, SJF. The global scheduler has a number of functions. One of these is matching the resources requested by a job to those available in the participating clusters. Another is to obtain the best utilization of the available clusters. The local scheduler is responsible for scheduling jobs to a specific resource. At both levels, the schedulers strive to maintain a good load balance.

Advantages:

a) It tries to reduce overall turnaround time and maximize system utilization for high system loads.
b) For high system loads it uses multi queue to maintain the delay of job scheduling at global level.
Disadvantages:
a) There may be a chance of underutilization of grid resources.
b) This algorithm does not consider the dynamic behavior of the grid resources.

III. Comparison of various grid scheduling algorithms

The following table shows the comparison of various grid scheduling algorithms. The basic parameters used for comparison are response time, resource utilization and load balance. Abbreviations used are

D-Distributed, H-Hierarchical, C-Centralized, HO-Homogenous, HE-Heterogeneous,
RT-Response Time, RU- Resource Utilization, LB-Load Balance, DY-Dynamicity, HI-High,
AVG-Average, LO-Low

Algorithm	Parameters					
	Architecture H/D/C	**Environment HE/HO**	**RT**	**RU**	**LB**	**DY**
Min-Min	**D**	**HE**	**HI**	**HI**	**AVG**	**HI**
Sufferage	**D**	**HE**	**HI**	**HI**	**AVG**	**HI**
HEFT	**D**	**HE**	**LO**	**HI**	**HI**	**HI**
COPS	**D**	**HE**	**LO**	**HI**	**HI**	**HI**
RASD	**D**	**HE**	**LO**	**HI**	**HI**	**HI**
HJS	**H**	**HO**	**AVG**	**HI**	**HI**	**HI**

Fig.2 Comparison of Different Grid Scheduling Algorithms

IV. Conclusion

In this paper we have discussed the different types of grid scheduling approaches and grid scheduling algorithms. The different grid scheduling algorithms are compared based on various factors like response time, resource utilization, load balancing etc.

References

[1]. N. Malarvizhi and V.R.Uthariaraj, Hierarchical Load Balancing Scheme for Computational Intensive Jobs in Grid Computing Environment, ICAC, IEEE, 2009

[2]. Grid Computing: Various Job Scheduling Strategies Abhang Swati Ashok Durole Pankaj Hari

[3]. Heuristics in Grid Scheduling, D. Thilagavathi and Dr. Antony SelvadossThanamani

[4]. Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, PrachetBhuyan "A Survey of Job Scheduling and Resource Management in Grid Computing", 2010.

[5]. Study of an Iterative Technique to Minimize Completion Times of Non-Makespan Machines, by Luis Diego Briceno, Mohana Oltikar, Howard Jay Siegel, and Anthony A. Maciejewski, 2007

[6]. M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, Dynamic mapping of a class of independent tasks onto heterogeneous computing systems, J. Parallel Distrib. Comput. 59, 2(Nov. 1999), 107_121.

[7]. Reliability-aware scheduling strategy for heterogeneous distributed computing systems XiaoyongTanga, Kenli Li , Renfa Li, BharadwajVeeravalli

[8]. Topcuoglu H, Hariri S, Wu M -Y. Performance-effective andlow complexity task scheduling for heterogeneous computing,IEEE Trans Parallel DistribSyst, 2002, 13(3): 260–274

[9]. G.Q. Liu, K.L. Poh, M. Xie, Iterative list scheduling for heterogeneous computing, J. Parallel Distrib. Comput. 65 (5) (2005) 654-665.