Task Scheduling Algorithms for Grid Computing with Static Jobs: A Review

Marish Kr. Singla
M.E. Scholar, Dept. of CSE
NITTTR, Chandigarh
profmarish@hotmail.com

Abstract: **Scheduling in traditional distributed systems has been mainly studied for system performance parameters without data transmission requirements. In task scheduling system for grid computing, most of algorithms such as priority based FCFS, reservation, backfilling etc. are commonly used for task scheduling. However, they have some inadequacies and adequacies in solving resources slot. In order to improve the system resource utilization rate and shorten the communication time, this paper describe the various task scheduling algorithm such as PB-FCFS and backfilling, Session Scheduling, Constraint-Based strategy for grid computing.**

Keywords: Task Scheduling, PB-FCFS with Backfilling, Session and Constraints based scheduling

1. INTRODUCTION

Grid computing is the novel technology for building high speed computing environment in which heterogeneous, homogenous, distributed and dynamically resources are integrated. It is different from the other distributed shared platforms, in which various heterogeneous and distributed resources are integrated and abstracted for higher level sharing. It is a challengeable question for efficient task scheduling algorithm design and implementation in grid environment [4]. Now a days, task scheduling problem becomes the field for research and provokes great attention from researchers [5, 6]

In order to find the suitable resources for task better, researchers have given many scheduling algorithm to accomplished the task, such algorithms including SJF, FCFS, Earliest Release Date(MRD), Minimum Time to Due Date(MTTD). These algorithms allot the task dynamically to the resources when the resource is relatively free. The allocation depends upon the task priority, static and dynamic creation of a task. The task may be categorized in two ways: dependent and independent task. Like this, the scheduling process may be divided into two categories such as task dependent and independent scheduling.

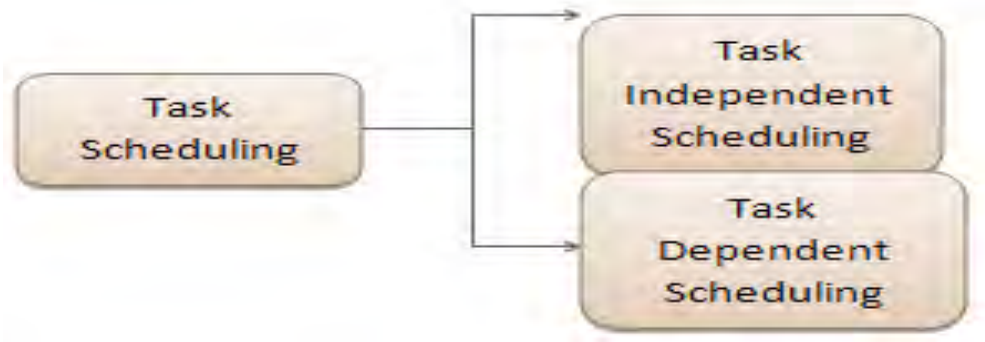

Figure 1. Task Scheduling

Scheduling process maps the task to resources available for task completion according to the specified set of constraints. Scheduling the independent task is to reduce the computational time. Each independent task is scheduled to available suitable resources. When tasks are dependent to each other's. Then, another task required the result of previous tasks to compute it. FCFS is adequate for dependent task. This paper describes the various task scheduling strategies and algorithm used in grid environment.

1. MODEL FOR TASK SCHEDULING

1.1 Analysis of Traditional Task Scheduling

A resource may satisfy the requirement of more than one task, and one task is possible to be matched by more than one resource [6]. The basic of task scheduling to selecting the best resource for appropriate task and analyzing the resource utilization rate, response rate, system efficiency etc. in task scheduling.

Figure2. Structure of traditional task scheduling model

The resource is divided according to the type of task it accomplished and the size of resources. Depend upon; the task is assigned to resources. How does the task matched the resource from pool of resources depends on their need?

1. The task published the resource information such as task type, size of resources and so on.
2. The resource will accurately reflect information back to task.
3. The task confirms information of resource.

1.2 Priority Based First Come First Serve (PB- FCFS)

On the basis of structure in Figure 1, the models structure adopts the multi-level updating strategy. The PB-FCFS task scheduling model integrates the advantages of priority and backfilling, etc. and expects to relieve resource slot to enhance the resource utilization rate. There are three restrictions for the PBFCFS task scheduling model:

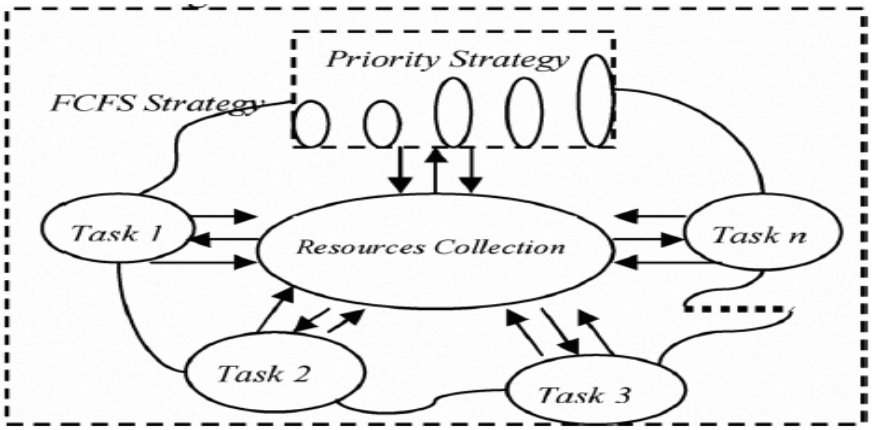

Figure3: Structure of PB-FCFS task scheduling model

Firstly, task will be selected and run orderly by FCFS strategy when entering the scheduling center forthe first time. If it does not have sufficient resource and task priority level, the task priority level is set as initial value by the scheduler, otherwise increase task priority level dynamically until the highest priority level.

Secondly, the remaining tasks will re-enter the loop scheduling process when a task implement is accomplished. Firstly, the task is selected and run orderly in terms of priority level. For tasks with the same priority level, the selection and implement of one task is according to FCFS strategy. Otherwise the scheduler will continue to increase task priority level until the highest priority is reached.

Finally, if there is insufficient resource, the first task in waiting queue must wait for the accomplishment of tasks in running queue and as well release resource. Immediately, the backfilling strategy is applied to insert some tasks with fewer resource requirements from waiting queue into the running queue.

The PB-FCFS algorithm introduces the ideas of priority and backfilling, so that it may improve the resource utilization rate effectively and shorten the response time to a certain extent. But, there should be resource reservation algorithm for the purpose of better resource allocation in terms of task information, and expand the definition of task priority so as to serve the system more preferably.

1.3 Session Scheduling Algorithm

The design goal of dynamic priority based proportional pattern is : each session is possible to get computing resource; in circs of the available resource can meet all the sessions, resource can be allocated to sessions in term of a proportion based on priority; in circs of resource is not sufficient, precede the high priority sessions take resource[5]. Under this pattern, resource is allocated based on priority. If priority is changed during runtime, system will reallocate computing resource. System executes the deserved logic core of each session every 500ms and dynamically adjusts the service instance distribution among sessions based on session priority and real-time running status.

Reservation pattern guarantees current sessions get minimum service instance. If request of minimum service instance from a session is fulfilled, this service instance will not be allocated to any other sessions. If available resource amount is less than sum of minimum resource amount requested, then minimum resource will allocated using priority-based FCFS strategy. If available resource amount is larger than the sum of minimum resource amount requested, minimum resource will be allocated using FCFS strategy, and then the remaining resource will be allocated based on session priority.

In order to overcome the low throughput of priority based FCFS strategy, Session scheduling represents two improvements: dynamic priority based proportional patter and reservation pattern.

2. CONSTRAINT-BASED JOB AND RESOURCE SCHEDULING

The Grid system is hierarchical two level systems. In user level, Grid user submits jobs to scheduler onto the grid environment. The scheduler obtains the information about the resources and its characteristics and sends jobs to the selected resources for execution.

A. Resource scheduling model

In resource scheduling, resources are arranged in hierarchical based shown in figure 4, where Heap Sort Tree (HST) is used to obtain the highest computational power resource or root node, so as to make balanced and effective job scheduling. Heap Sort Tree is well suitable for the complex grid environment because it is very effective in large scale sorting applications. When the resources join into the grid, they are arranged in a tree by Heap sort Tree using their computational power. The root node of the tree

having highest computational power in whole grid system is ready to compute the jobs.

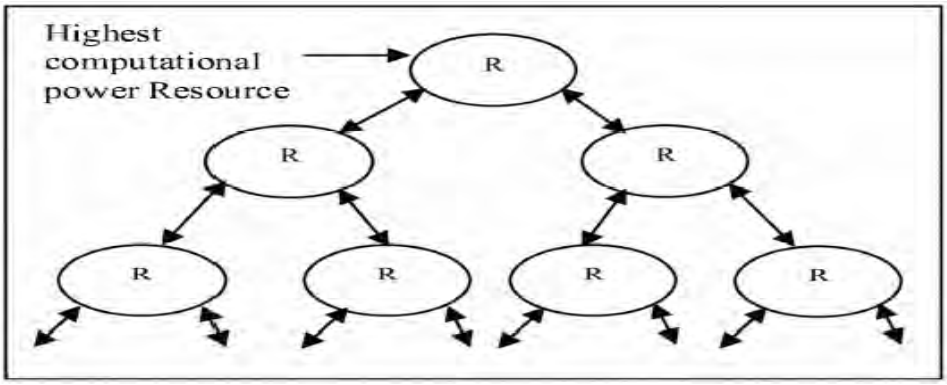

Figure4: Highest Computational power resource selection [4]

B. Job Scheduling

Job scheduling is the mapping of jobs to specific physical resources but, assigning of a single job to the specific resource taking a high processing time and communication time. So, processing and communication time can be reduced considering grouping strategy.

This Grouping strategy is based on processing capability (in MIPS), bandwidth (in Mb/s), and memory-size (in Mb) of the available resources. Jobs are put into the job group until all the following conditions are satisfied.

Condition 1: Groupedjob_MI <::: Resource_MIPS * Granularity size

Condition 2: Groupedjob _ MS <::: Resource _ MS

Condition 3: Groupedjob_MS <::: Resource_baudJate * Tcomm

Constraint-Based Job and Resource scheduling algorithm has been proposed taking three constraints into account.

3. LOAD BALANCING STRATEGIES FOR GRID COMPUTING

A typical distributed system will have a number of interconnected resources who can work independently or in cooperation with each other. Each resource has owner workload, which represents an amount of work to be performed and every one may have a different processing capability.

The static load balancing problem for a mesh based application involves partitioning into subdomains. The subdomains can then be distributed over the processors and calculation carried out in parallel.

Dynamic Load Balancing (DLB) is used to provide application level load balancing for individual parallel jobs. It ensures that all loads submitted through the DLB environment are distributed in such a way that the overall load in the system is balanced and application programs get maximum benefit from available resources. Current version of the DLB has two major parts. One is called System Agent that collects system related information such as load of the system and the communication latency between computers. The other is called DLB Agent which is responsible to perform the load balancing.

After load balancing at group level (if load is not balanced at group level) it will be balanced at region level and after balancing at region level (if load is not balanced at region level) it will be balanced at grid level.

CONCLUSION

This paper shows taxonomy of task-aware scheduling based on a set of requirements such data transmission, decoupling of task from processing, and load balancing strategies. By considering these requirements a family of tasks scheduling problems can be defined, whose resolution can be very useful to design efficient data-aware schedulers. We have focused on the task independent batch task scheduling for which we have formalized the transmission time, in a way that it can be easily integrated into classical optimization objectives of Grid scheduling.

REFERENCES

[1] Wanneng Shu, Shijue Zheng, Wei Ma, Guangdong Chen,Jianhua Du, “A Grid Computing Task Scheduling Method Based on Target Genetic Algorithm” 6th World Congress on Intelligent Control

[2] Hong Jiang, Tianwei NI, “PB-FCFS--A Task Scheduling Algorithm Based on FCFS and Backfilling Strategy for Grid Computing”, IEEE, 978-1-4244-5228-6/09

[3] Vishnu Kant Soni , Raksha Sharma, “Constraint-Based Job and Resource scheduling in Grid Computing”, IEEE- 978-1-4244-5540-9/10

[4] S. Gomathi, “An adaptive grouping based Job Scheduling in Grid Computing”, ICSCCN 2011

[5] Sha Fan, 2010, “Session Scheduling Algorithm of Grid Computing”, Third International Conference on Knowledge Discovery and Data Mining,

[6] Jagdish Chandra Patni, Dr. M.S.Aswal, 2011, “Load balancing Strategies for Grid Computing”