

# A study on Task Scheduling Algorithms in Grids

W.Viyakulamary<sup>1</sup>, T.Kokilavani<sup>2</sup>,

<sup>1</sup>Research Scholar, Department of Computer Science, St.Joseph's College (Autonomous),  
Tiruchirapalli-620 002.

E-mail : [viyakulamary14@gmail.com](mailto:viyakulamary14@gmail.com)

<sup>2</sup>Assistant Professor, Department of Computer Science, St.Joseph's College (Autonomous),  
Tiruchirapalli-620 002.

E-mail : [yani78\\_ram@yahoo.com](mailto:yani78_ram@yahoo.com)

**Abstract:** Grid computing is the collaboration of interconnected computing resources ready to be served for the user's request for complex and time consuming processes. Task scheduling is one among the thrust research areas of grid computing as it can be viewed as an NP-complete problem. Task scheduling is still complicated as the resources in the grid environment have unique characteristics in nature. Thus, the allocations of resources have limited opportunities in finding out the optimal solution. Over the past decades, many researches have been proposed on heuristics task scheduling algorithms that contributed a substantial impact on the performance of task scheduling. Unfortunately, the algorithms are still lacking in producing cent percent of optimal solution in the allocation of resources to the needy systems. The evaluations of the optimal solution are also found more difficult to prove the efficiency of the algorithms. Therefore, this review is motivated to present the depth study on grid computing environment, existing heuristics scheduling algorithms and their significances. This paper also includes the comparative analysis of the reviewed algorithms using the most dominating parameters.

**Keywords:** grid, tasks, resources, NP-complete, Heuristics

## I.Introduction

The remarkable enhancements of network technologies have led us into new era of performing high computational tasks with the help of the resources available in the high speed networked environment. Foster[1] defined, grid computing as a collection of large number of heterogeneous, distributed, decentralized and dynamic resources. Grid resources enhance the computational speed by splitting up the task into smaller chunks and performed by numerous computational tasks owned by the final public for the faster the completion of task. The resources are distributed over multiple controlled domains that are executed in large scale applications [2]. Appropriate scheduling of the grid resources helps in achieving the target of the grid computing.

Grid computing can be viewed a process of distributed computing that shares data, application, computing and networking resources and storage [3]. Grid computing promises to deal complex computational issues, storage capacity and network information measure by granting users a wider IT capability. Grid application starts by meeting out the collection of customer needs followed by selecting the appropriate grid computing resources catering to the needs.



**Figure 1:** Heterogeneous Resources of Grid Computing

Grid resources optimally utilize the CPU cycles, that wouldn't make the resources go wasted. Moreover, users get add on computational resources that could process large-scale computational tasks as equal to supercomputer.

### Advantages of Grid Computing

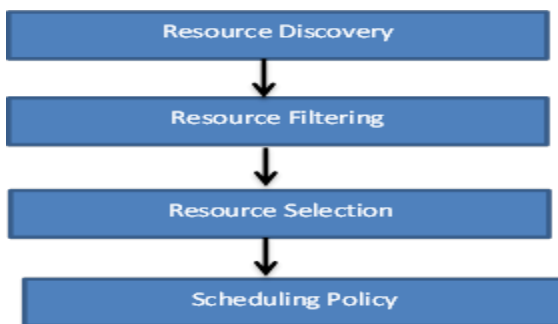
- Saves financial resources
- Solves larger and complex problems within short time
- uses idle resources
- less vulnerable for failure
- Manages grid software policies for effective utilization
- Executes parallel tasks for speedy performance

## Disadvantages of Grid Computing

- Need to have fast net connection
- Licensing of server prohibits grid apps
- Contains small servers for big administrative domain
- Political challenges associated with sharing of resources

## II.Task Scheduling in Grid

Task scheduling refers to the way in which the processes are assigned to run on the available processors [4]. Task Scheduling is concerned with throughput, waiting time, processor utilization, turn around, response time and most importantly the minimization of make span. The process of task scheduling in Grid cannot be done in a single step; rather, it has to process several steps before allocating the tasks to the resources. Fig. 2 shows the various phases involved in task scheduling process [5].



**Figure 2:** Phases of Task Scheduling

**Resource Discovery:** maintains the status information of the resources and identifies the geographical locations of authorized and reliable resources

**Resource Filtering :** This phase is meant for resource elimination. The resources which do not meet out the requirements specified by the user are eliminated. Those resources which are chosen to be allocated to task alone travels to the next phase.

**Resource Selection :** The goal of this phase is to choose the appropriate resource among the filtered resources. The selection phase requires the detailed dynamic status information for proper ranking of those resources, which most likely to satisfy the scheduling objectives.

**Scheduling Policies:** This phase defines the strategy for mapping and submission of tasks to the resources.

- **Tasks Mapping:** Here, tasks are allocated to the selected resources based on some scheduling strategy.
- **Tasks Submission and Gathering of Computed results:** This is the last step of common tasks scheduling process. In this phase, dispatcher submits the tasks to the respective resources and keeps on monitoring till its completion. The

computational results are collected from the resources through the output receiver.

## III.Review on Traditional Task Scheduling Algorithm

The task scheduling is the process of assigning the tasks in a machine to the resources of grid in a manner that will optimize the overall performance of the application, while assuring the correctness of the result. The primary goal of task scheduling is to schedule tasks on processors and minimize the makespan of the schedule, i.e., the completion time of the last task relative to the start time of the first task. The output of the problem is an assignment of tasks to processors.

The objective of this paper is to review the existing task scheduling algorithms in grid computing.

### Opportunistic Load Balancing

OLB allocates the tasks in random order to the next available machine irrespective of the task's expected execution time on the system. OLB aims to balance the load among the Grid system machines but it leads to poor makespan [6]. In this method earliest machine which is idle is selected without considering the task's execution time. If two or more machines are idle then machine is selected arbitrarily. In this method time required for Scheduling is less and it keeps almost all the machines busy at all possible time.

### Minimum Execution Time

MET algorithm finds the task which has minimum execution time and assigns the task to the resource based on first come first served basis. The limitation of this algorithm is load imbalance. It does not consider the availability of the resource and its load. MET assigns each task in arbitrary order to the machine with the minimum expected execution time for the task. MET assigns each task its best machine[4].

### Minimum Completion Time

MCT algorithm finds the machine which has Minimum Completion Time for the particular task. It assigns the task to resources based on completion time. Completion time is calculated by adding the execution time and the ready time of the resource [7]. MCT assigns each task in arbitrary order to the machine with the minimum completion time for the task. Thus, some tasks may not assign to the machine with minimum execution time. The assigned task is deleted from the set of tasks, and the completion times for all the remaining tasks are updated. This process continues until all tasks are mapped

### Min-Min

Min-min algorithm first computes the completion time for each task on each machine. Then, the machine with the minimum completion time for each task is selected. Finally, it maps the task with the minimum completion time to the selected machine. The assigned task is deleted from the set

of tasks, and the completion times for all the remaining tasks are updated. This process is repeated until all tasks are mapped [7]. Min-Min aims to minimize the makespan by assigning tasks with minimum completion time (small tasks) to the faster machines first followed by the tasks with longer completion time (large tasks). This results in a load unbalance and poor utilization.

### Max-Min

Max-min first computes the completion time for each task on each machine. Then, the machine with the minimum completion time for each task is selected. Finally, it maps the task with the maximum completion time to the selected machine. The assigned task is deleted from the set of tasks, and the completion times for all the remaining tasks are updated. This process continues until all tasks are mapped [8]. Max-min achieves better performance better than the Min-min algorithm when the number of small tasks is larger than the number of long tasks.

### Round Robin Technique

Chang et al [9] introduced RoundRobin (RR) algorithm that used ring as its queue to store tasks. The tasks which have same execution time are clubbed into the queue and waiting for their turn. Each task is executed simultaneously till their turn getting finished. So, the tasks can no longer wait for other tasks to be completed since, all the tasks are focused simultaneously. But if the load is found to be heavy, RR will be taking a very long time to complete all the tasks. Each task will be given a priority value by the priority scheduling algorithm and the same is used to dispatch tasks. The priority value of each task will be depending on the

tasks status such as the requirement of memory sizes, CPU time and so on. The main problem of this algorithm is that it may be causing an indefinite blocking or starvation if the requirement of a task is never being satisfied.

### First-Come, First-Served Scheduling

One of the simplest task scheduling algorithms is First-Come, First-Served Scheduling which is based on the theory that the resources that are arrived first will be focused first and so on. Therefore, the algorithm uses the order of executed tasks in the order of submission. If a significant amount of time is essential to complete the first task, the same amount of time is used up waiting for the execution of the next task. As a result, a convey result is created and the whole system performance will be reduced [10].

### Sufferage

The task's sufferage value is the difference between its best minimum completion time and its second best minimum completion time. The task with high sufferage value is selected and mapped to the machine with the minimum completion time [4]. The assigned task is deleted from the set of tasks, and the completion times for all the remaining tasks are updated. This process is repeated until all tasks are mapped. Since the Sufferage heuristic schedule the task that would highly suffer if it is not assigned to the machine with the minimum completion time first, it is expected to perform well in the systems with machines are highly variant in their execution times for a specific task. Table 1 gives the summary on the traditional task scheduling algorithms with their merits and demerits [11][4].

**Table 1:** Summary on Traditional Task Scheduling Algorithms

S.No	Algorithm	Merits	Demerits
1	Opportunistic Load Balancing(OLB)	<ul style="list-style-type: none"> <li>• Minimum make span</li> <li>• Maximum resource utilization</li> <li>• Low computational complexity</li> </ul>	<ul style="list-style-type: none"> <li>• Flow Time was not considered</li> </ul>
2	Minimum Execution Time(MET)	<ul style="list-style-type: none"> <li>• Each task is assigned with the best machine in terms of minimum execution time in arbitrary order</li> </ul>	<ul style="list-style-type: none"> <li>• Causes severe load imbalance across machines</li> <li>• Tasks are assigned without considering the availability of resources</li> </ul>
3	Minimum Completion Time(MCT)	<ul style="list-style-type: none"> <li>• Tasks are assigned to the machines with the minimum expected completion time for the task</li> <li>• Combines the benefits of load balancing and minimum execution time</li> </ul>	<ul style="list-style-type: none"> <li>• Causes , the tasks to be assigned to machines that do not have minimum execution time</li> </ul>
4	Min-Min	<ul style="list-style-type: none"> <li>• Considers all unmapped tasks during mapping</li> <li>• Considers the availability of machines</li> </ul>	<ul style="list-style-type: none"> <li>• The percentage of tasks assigned to</li> <li>• their first choice is likely to be very high which may lead small make span</li> </ul>

5	Max-Min	<ul style="list-style-type: none"> <li>Minimizes the penalty incurred from performing tasks with longer execution times</li> <li>Prevents processors stay idle for long time</li> <li>More balanced mapping with better make span</li> </ul>	<ul style="list-style-type: none"> <li>Flow time is not focused</li> </ul>
6	Round –Robin	<ul style="list-style-type: none"> <li>Tasks do not have to be waited for other tasks to be completed</li> <li>Simultaneous process jobs minimizes the completion time</li> </ul>	<ul style="list-style-type: none"> <li>Takes long time to complete execution</li> <li>May cause indefinite blocking or starvation</li> </ul>
7	First come First Server	<ul style="list-style-type: none"> <li>Simple</li> <li>Minimizes the waiting time</li> </ul>	<ul style="list-style-type: none"> <li>poor make span</li> <li>not optimal</li> </ul>
8	Sufferage	<ul style="list-style-type: none"> <li>performs well with the systems that are highly with variant completion time</li> </ul>	<ul style="list-style-type: none"> <li>highly suffers if it is not assigned to the machine with the minimum completion time</li> </ul>

#### IV. Conclusion

Task scheduling is one of the critical issues in grid computing. Scheduling often challenges grid computing environment in allocating suitable resource for the given task. In the recent past, many task scheduling algorithms have been proposed for the effective utilization of grid resources for reducing the overall completion time. The main objective of task scheduling is to increase the throughput based on availability of resources. In this paper five different traditional grid task scheduling algorithms and their working principle are taken for the survey such as MCT, MET, Min-Min, Max-Min and Sufferage. Though, the algorithms are said to be natural, each has its own merits and demerits in the process of task scheduling. In future these algorithms can be combined with each other for increasing the accuracy of resource allocation.

#### References

- Ian Foster and Carl Kesselman, "The Grid 2: Blueprint for a New Computing Infrastructure", Second Edition, Morgan Kaufmann, 2004.
- Elwyn Betzar, Arul Xavier .V.M, "Survey On Heuristics Based Resource Scheduling In Grid Computing", Indian Journal of Computer Science and Engineering (IJCS), Vol. 5 No.1 Feb-Mar 2014.
- S.GokulDev, R.Lalith Kumar, "Tasks Scheduling in Grid Computing using User Deadline", International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 1387-1392, 2014.
- R. Venkatesan, J. Raj Thilak, "Perspective Study on task scheduling in computational grid", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 1, Issue 8, October 2012.
- Manoj Kumar Mishra, Yashwant Singh Patel, Yajnaseni Rout, G.B. Mund, "A Survey on Scheduling Heuristics in Grid Computing Environment", I.J. Modern Education and Computer Science, 2014
- Seema Kumari, Gulshan Kumar, "Survey on Tasks Scheduling Algorithms in Grid Computing", International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 15, April 2015.
- M. Hemamalini, "Review on Grid Task Scheduling in Distributed Heterogeneous Environment", International Journal of Computer Applications (0975 – 8887), Volume 40– No.2, February 2012.
- Fahd Alharbi, "Simple Scheduling Algorithm with Load Balancing for Grid Computing", Asian Transactions on Computers (ATC ISSN: 2221-4275) Volume 02 Issue 02, 2012.
- Ruay-Shiung Chang, Jih-Sheng Chang, Po-Sheng Lin, "An ant algorithm for balanced tasks scheduling in grids", Future Generation Computer Systems 25(2009) 20–27.
- S. Gokuldev, Lija Anna Mathew, "Task Graph Based Scheduling And Resource Monitoring In Grid Computing", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 5, May – 2013 ISSN: 2278-0181
- Dr. D.I. George Amalarethnam, P. Muthulakshmi, "An Overview of the Scheduling Policies and Algorithms in Grid Computing", International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 2, No. 2, April 2011