IJCTA, 9(27), 2016, pp. 71-76 ISSN: 0974-5572
© International Science Press

Reduced Makespan Task Scheduling Algorithm for Grid Computing

T. Kokilavani* **and D.I. George Amalarethinam****

ABSTRACT

Grid Computing is used to solve computation intensive large scientific problems which cannot be solved using a single computer. Task scheduling is an important factor in computational grids. Users from various geographical locations submit their tasks to be executed on the grid resources. This presents the challenge to schedule these tasks in shared heterogeneous systems. An efficient task scheduling algorithm maximizes the performance of Grid computing. Many researchers focus on task scheduling which is NP-complete problem. The popular Min-min algorithm schedules the smaller tasks first which makes larger tasks to wait for more time. In this paper, a new task scheduling algorithm RMTSA (Reduced Makespan Task Scheduling Algorithm) is proposed which schedules the smaller tasks and larger tasks alternatively. This overcomes the limitation of Min-min algorithm. RMTSA switches between the smaller and larger tasks which reduces the makespan as well as maximizes the resource utilization.

Keywords: Grid computing, Task scheduling, Resource utilization, Min-Min algorithm.

1. INTRODUCTION

Grid computing has become a new form of conventional distributed computing because of its focus on large-scale resource sharing and high-performance orientation. Sharing in grid computing is not file exchange rather it is direct access to computers, software, data and other resources. The resource providers and consumers must clearly define what is shared, who is sharing and the conditions under which sharing can occur. Such set of rules among individuals and institutions is called Virtual organization [1]. The unique nature of grid is that the resources are geographically distributed, that is they may belong to different administrative domains like universities, organizations and different countries [2]. Grid involves the collaborative use of scientific instruments, networks, databases and computers owned and managed by different organizations. In a grid environment the end users submit their jobs to the Grid Resource Broker. The resource providers register their resource characteristics within one or more Grid Information Services. The scheduler in Grid Resource Broker identifies appropriate resources by querying the information services, and schedules the application jobs for execution on those resources. Finally, the resource broker is responsible for monitoring the execution of jobs until they are completed [3].

Scheduling is a process that assigns jobs to available resources based on user objective functions. Scheduling has been extensively studied with respect to traditional, single and distributed computing systems before Grid systems became apparent. The phases of common job scheduling process are Resource Discovery, Resource Filtering, Resource Selection and Scheduling Policy[4][5]. In general, scheduling problems are proved to be NP-Complete [6]. But Grid computing has its own difficulties because of the heterogeneity and availability of resources. In this paper Reduced Makespan Task Scheduling Algorithm (RMTSA) is proposed which minimizes the makespan and maximizes the resource utilization.

* Assistant Professor, Department of Computer Science, St. Joseph's College, Trichy, *Email: vani78_ram@yahoo.com*

** Dean of Science & Director-MCA, Jamal Mohamed College, Trichy, *Email: di_george@ymail.com*

2. LITERATURE REVIEW

Task scheduling is a challenging problem in grid computing environment [7]. Task scheduling policies applied for dedicated systems cannot be applied for non-dedicated systems like Grid Computing. So task scheduling in grid computing which is one of NP-Complete problems becomes a focus of many scholars in grid computing area. A job can be subdivided into tasks for parallel execution. The tasks can be dependent or independent in nature. Non-communicating independent tasks called as metatasks occur in many situations. An example of independent tasks would be an image processing application working on a group of different set of pixels. In independent task scheduling, there is no dependency among the tasks and so the tasks may be scheduled in any order. For independent task scheduling, researchers use Expected Time to Compute (ETC) matrix to select a resource for a particular task.

Braun et al., [8] studied the relative performance of eleven mapping heuristics by varying the basic characteristics of the Heterogeneous Computing environments. The performance of these heuristics were studied by varying both task and resource heterogeneity. It was assumed that the heuristics derive a mapping statically (i.e., off-line). It was also assumed that a metatask is being mapped and that the goal is to minimize the total execution time of the metatask. The eleven heuristics examined were Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-Min, Duplex, Genetic Algorithm (GA), Simulated Annealing, Genetic Simulated Annealing, Tabu, and A*. For the situations, implementations, and parameter values used, GA consistently gave the best results. The simple Min-Min heuristic was next to GA.

Seema kumari et al [9] have surveyed various job scheduling algorithm in grid computing that will benefit for the researchers to carry out the future work in that area and develop a better algorithm.

Keerthika et al [10] presents a new algorithm that mainly focuses on better meeting the deadlines of the statically available jobs as expected by the users. This algorithm also concentrates on the better utilization of the available heterogeneous resources.

Shanthini et al [11] have proposed BG_ATC algorithm which is a combination of best gap (BG) search and Apparent Tardiness Cost (ATC) indexing algorithm. In this work, they have used best gap at resource level so there is only minor improvisation in the makespan compared with close competitive algorithms.

Doreen. D et al., [12] have proposed an efficient Double Min-min Algorithm which performs scheduling in order to enhance system performance in Hypercubic P2P Grid (HPGRID). The results shows that the SPA (Set Pair Analysis) based Double Min-min scheduling minimizes the makespan with load balancing and guarantees the high system availability in system performance.

He. X et al., [13] have presented a QoS guided Min-Min task scheduling algorithm which schedules tasks requiring high bandwidth before the others. Therefore, if the bandwidth required by different tasks varies highly, the QoS guided Min-Min algorithm provides better results than the Min-Min algorithm. Whenever the bandwidth requirement of all of the tasks is almost the same, the QoS guided Min-Min algorithm acts similar to the Min-Min algorithm.

Kamalam et al., [14] have proposed a new task scheduling algorithm named Min-mean heuristic algorithm for static mapping to achieve better performance. The Min-mean algorithm reschedules the schedule produced by Min-min based on the mean makespan of all the resources. This algorithm does not produce a better schedule for highly heterogeneous resources.

Sameer Singh et al., [15] have proposed two heuristic algorithms: QoS Guided Weighted Mean Time-Min(QWMTM) and QoS Guided Weighted Mean Time Min-Min Max-Min Selective(QWMTS). These algorithms are proposed for batch mode independent tasks scheduling. The QoS parameter by these algorithms is the network bandwidth.

Soheil Anousha et al [16] have proposed An Improved Min-Min Task Scheduling Algorithm which computes average of completion time and standard deviation of existing tasks. The proposed algorithm compares values of

ACT and SD. If ACT is less than SD, it will select from front of queue to assign the next task. Otherwise, it will select from rear of queue to assign the next task.

Saeed Parsa et al [17] have proposed RASA, a new task scheduling algorithm which applies Min-min strategy to assign the first task if the number of available resources is odd, otherwise the Max-min strategy is applied.

Improved Min-Min Task Scheduling Algorithm and RASA choose the smaller tasks or larger tasks based on constraints which will not give equal opportunity to all tasks. The proposed RMTSA algorithm chooses smaller tasks and larger tasks alternatively without any constraint. This gives equal opportunity to both type of tasks which reduces the makespan and maximizes the resource utilization.

3. PROPOSED RMTSA ALGORITHM

The main objective of any scheduling algorithm is to minimize the makespan and it can be calculated as follows:

$$\text{makespan} = \max(CT(Ti, Mj)) \tag{1}$$

$$CTij = Rj + ETij \tag{2}$$

where CT is the completion time of a resource, Rj is the ready time of a resource and ETij is the execution time of task *i* on resource *j* which is provided by ETC (Expected Time to Compute) matrix.

The traditional min-min algorithm schedules the smaller tasks first which increases the wait time of longer tasks. Max-min algorithm schedules the longer tasks first. But, if the number of large tasks increases the short tasks suffer in max-min algorithm. To overcome the limitations of both min-min and max-min algorithm Reduced Makespan Task Scheduling Algorithm (RMTSA) is proposed in this paper. In RMTSA, the minimum execution time of each task is identified initially. Then the minimum execution times are sorted in ascending order and placed in a queue. This places the short tasks in the front end of the queue and large tasks in the rear end of the queue. Then the tasks from front end of the queue and rear end of the queue are taken alternatively for scheduling. The tasks are scheduled based on Minimum Completion Time (MCT). This gives equal opportunity to both short and large tasks which minimizes the makespan. In addition to minimizing the makespan RMTSA maximizes the resource utilization. The Pseudocode for RMTSA is given in figure 1.

```
Find the minimum execution time (MET) of each task.
Sort these METs in ascending order
Put this sorted list in a queue.
Front = 0
For all tasks
Begin
    If front = 0
    Begin
        Select the task from front of the queue and schedule based on MCT.
        Set front =1
    End
    Else if front = 1
    Begin
        Select the task from rear of the queue and schedule based on MCT.
        Set front = 0
    End
End
```

Figure 1. Pseudocode of RMTSA

3.1. An Illustrative Example

For simplicity, 3 tasks and 2 resources are considered. The Expected Time to Compute (ETC) matrix for these tasks is given in Table 1.

Table 1
Sample ETC Matrix

Tasks	*Resources*	
	R1	*R2*
T1	2	3
T2	3	6
T3	6	12

Resource R1 produces minimum completion time for all resources. So Min-min algorithm schedules all the tasks in R1 which makes R2 idle. RMTSA initially schedules task T1, then schedules task T3 and finally task T2.

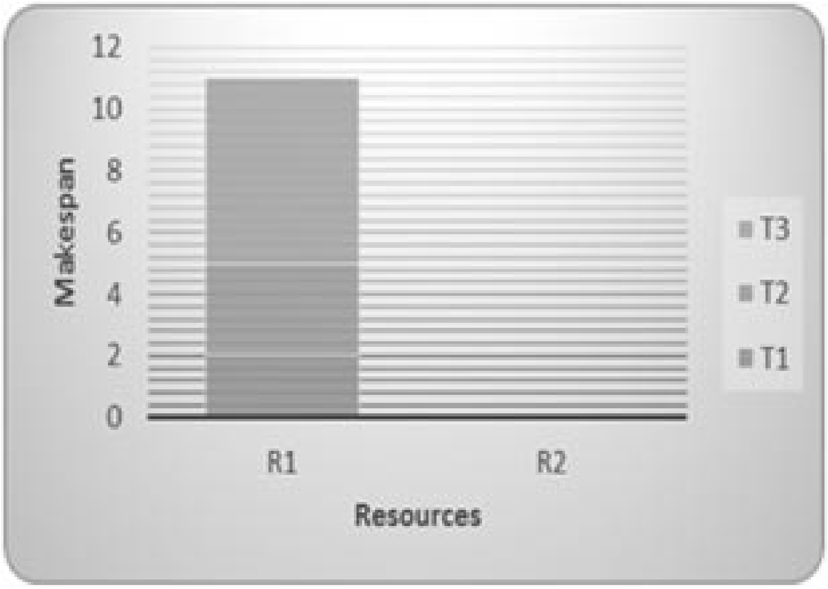

Figure 2: Min-min algorithm

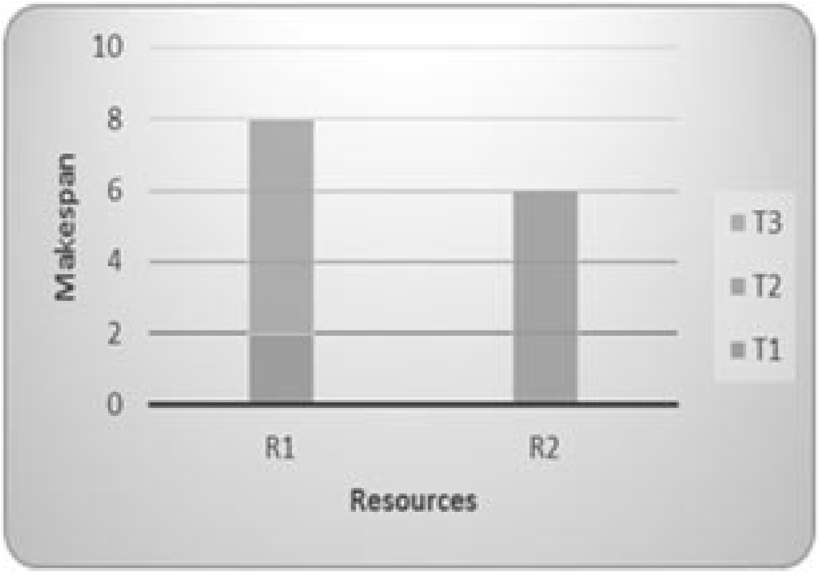

Figure 3: RMTSA algorithm

While scheduling T2, resource R2 gives minimum completion time. So RMTSA schedules task T2 in resource R2. Thus RMTSA effectively utilizes all resources while minimizing the makespan. Applying the data given in Table 1, the Min-Min algorithm gives a makespan of 11 whereas proposed RMTSA gives a makespan of 8 time units. Figure 2 gives the result of applying min-min algorithm and figure 3 gives the result of applying RMTSA algorithm.

4. EXPERIMENTS AND RESULTS

To evaluate the performance of proposed algorithm experiments are conducted by varying ETC matrices. The experiments are conducted with twelve types of ETC matrices generated by EMGEN tool [18]. Three different types of ETC matrices are generated based on consistency.

The resources considered here may be consistent, partially consistent or inconsistent. Various experiments are conducted with varied number of tasks and resources and in all cases RMTSA produces less makespan than Min-min algorithm. The results for 12 instances are shown in table 2. The instances use a 3 character code to give the type of task and resource. The first alphabet denotes whether the matrix is consistent/inconsistent/partially consistent.

Table 2
Comparison of Min-Min and RMTSA

INSTANCES	MIN-MIN	RMTSA
CLL	503	382
CLH	2265	1549
CHL	3275	2180
CHH	4166	3082
ICLL	358	316
ICLH	1341	1320
ICHL	2207	2112
ICHH	3674	3230
PCLL	344	339
PCLH	1478	1475
PCHL	1798	1670
PCHH	3864	3624

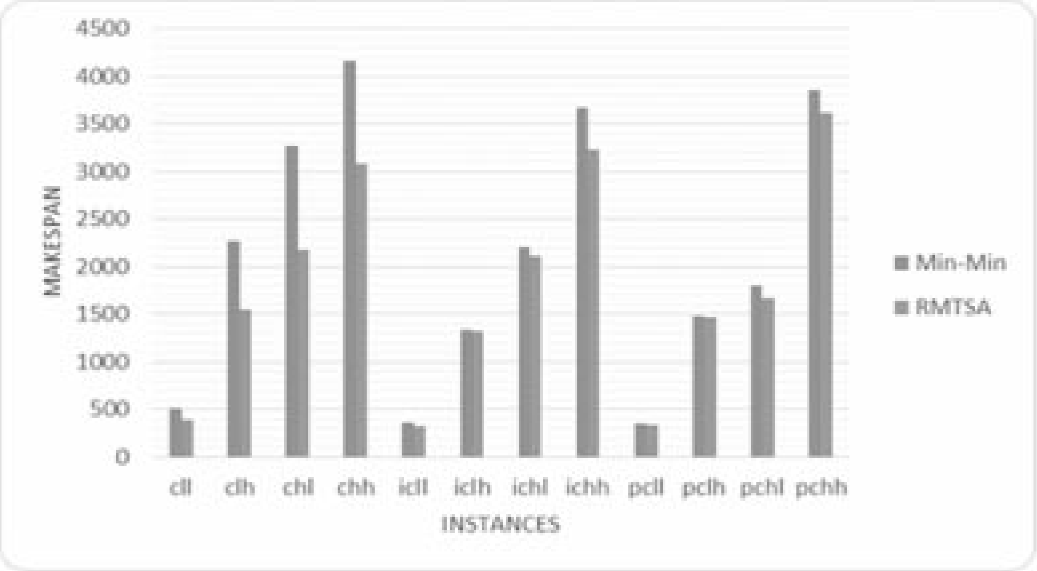

Figure 4: Makespan chart for Min-Min and RMTSA

The second alphabet denotes the heterogeneity of tasks [low/high]. The third alphabet denotes the heterogeneity of resources [low/high]. Figure 4 gives the graphical representation of the results.

5. CONCLUSION

Task Scheduling is an important factor in Grid Computing. Many metatask scheduling algorithms are available in the literature. But they fail to use the resources effectively. Min-Min algorithm schedules the smaller tasks first which makes large task to wait for more time. It also increases the makespan. Max-min algorithm chooses larger tasks initially. But, if the number of large tasks increases then smaller tasks will suffer. Reduced Makespan Task Scheduling Algorithm proposed in this paper alternatively schedules the smaller tasks and larger tasks which gives equal opportunity to both type of tasks. The experimental results obtained show that RMTSA outperforms the Min-min algorithm. RMTSA reduces the makespan while maximizing the resource utilization.

REFERENCES

[1] Ian Foster, Carl Kesselman, Steven Tuecke, "The Anatomy of the Grid Enabling Scalable Virtual Organizations", *International Journal of Supercomputer Applications*, 1 – 25, 2001.

[2] Michael Gerndt, "Grid Computing - An Infrastructure for Large Scale Simulations", 2004.

[3] Rajkumar Buyya, Sri kumar Venugopal, "A Gentle Introduction to Grid Computing and Technologies", *Computer Society of India*, 2005.

[4] Franke. C, U. Schwiegelshohn, R. Yahyapour, "Job Scheduling for Computational Grids", *University of Dortmund, Technical Report 0206*, 2006.

[5] Y. Zhu, "A Survey on Grid Scheduling Systems", *Department of Computer Science, Hong Kong University of science and Technology*, 2003.

[6] Fernandez-Baca D, "Allocating Modules to Processors in a Distributed System", *IEEE Transactions on Software Engineering*, **15(11),** 1427-1436, 1989.

[7] Ian Foster and Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", ISBN 1-55860-475-8, July 1998.

[8] Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M.Reuther, A.I., Robertson, J.P., et al. "A Comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", *Journal of Parallel and Distributed Computing*, **61(6)**, 810-837, 2001.

[9] Seema Kumari, Gulshan Kumar, "Survey on Job Scheduling Algorithms in Grid Computing", *International Journal of Computer Applications (0975 – 8887),* **115(15)**, April 2015.

[10] P. Keerthika, P.Suresh, "An Adaptive Algorithm for Task Scheduling For Computational Grid", *IJRET: International Journal of Research in Engineering and Technology*, **04(03)**, Mar-2015

[11] J. Shanthini, T. Kalaikumaran, and S. Karthik, "Hybrid Scheduling Model for Independent Grid Tasks", *Scientific World Journal*, 1 -9, September 2015,

[12] Doreen Hephzibah Miriam. D and Easwarakumar. K.S, "A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems", *IJCSI International Journal of Computer Science Issues*, **7(4)**, July 2010.

[13] He. X, X-He Sun, and Laszewski. G.V, "QoS Guided Minmin Heuristic for Grid Task Scheduling", *Journal of Computer Science and Technology*, **18**, 442-451, 2003.

[14] Kamalam.G.K and Muralibhaskaran.V, , "A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-Tasks On Heterogenous Computing Systems", *IJCSNS International Journal of Computer Science and Network Security*, **10(1)**, January 2010.

[15] Sameer Singh Chauhan,R. Joshi. C, "QoS Guided Heuristic Algorithms for Grid Task Scheduling", *International Journal of Computer Applications (0975 – 8887)*, **2(9)**, 24-31, June 2010.

[16] Soheil Anousha, Mahmoud Ahmadi, "An Improved Min-Min Task Scheduling Algorithm in Grid Computing", *LNCS 7861*, 103–113, 2013.

[17] Saeed Parsa and Reza Entezari-Maleki, "RASA: A New Task Scheduling Algorithm in Grid Environment", *World Applied Sciences Journal 7 (Special Issue of Computer & IT),* 152-160, 2009

[18] T. Kokilavani, Dr. D.I. George Amalarethinam, "EMGEN – A Tool To Create ETC matrix With Memory Characteristics For Meta Task Scheduling In Grid Environment", *The International Journal of Computer Science & Applications (TIJCSA)*, **2(03)**, 84 – 91, May 2013