International Journal of Communication and Networking System
Volume: 05 Issue: 02 December 2016,Pages No.96-100
ISSN: 2278-2427

Performance Analysis of Balanced Minimum Execution Time Grid Task scheduling Algorithm

M. Hemamalini[1], M.V. Srinath[2]
[1] Research scholar, STET Women's College, Sundarakkottai, Mannargudi, India.
[2] Director, Department of MCA, Sengamala Thayaar Educational Trust Women's College, Mannargudi, India.
Email-ID: maliniavcce@gmail.com, sri_induja@rediffmail.com

Abstract - Computational grid is the most promising technology generally used for distributed environment. The issues associated with Grid task scheduling are resource discovery, heterogeneity and fault tolerance. Efficient scheduling algorithm is essential for effective utilization of the resources and reduces the makespan. This paper shows the performance of Balanced Minimum Execution Time task scheduling algorithm with other algorithms such as Min-Min, Load Balanced Min-Min (LBMM), Minimum Execution Time based on execution time, makespan, completion time and load balancing. The BMET scheduling algorithm increases the resource utilization and the load is balanced by rescheduling the resources. This survey shows that Balanced Minimum Execution Time outperforms the Min-Min, LBMM and Minimum Execution Time task scheduling algorithms.

Keywords: Grid Computing, Task scheduling, Load balancing, Execution Time, Heterogeneous computing and Makespan.

I. INTRODUCTION

Grid computing technologies, particularly the ones used in High Throughput Computing (HTC) systems are used to solve large scale and complicated problems. Grid computing emphasizes coordinating and sharing of computational power and resources like data, storage, bandwidth and information. The main purpose of scheduling is to increase utilization of the resources. The key for these infrastructures to offer a high performance is the effective utilization of available computing resources [1]. The Grid architectures offer a middleware technology for allocation of resource, task scheduling, authorization, security and data management [2]. One of the important components of parallel and distributed computing is task scheduling. Task scheduling deals with policies, models, bandwidth, objective functions and dynamic behaviour of resources.

Several grid task scheduling algorithms are existing to reduce the makespan [3], [4], [5], [6], [7]. With the explosion of the Grid, there are two conceptions which needs to be considered in a scheduling model namely non-dedicated network and Quality of Service.

II. REVIEW OF LITERATURE

Nadar Mohammed et al., proposed delay tolerant load balanced task scheduling algorithm, in which the client is the central control of the process. It instructs the server to initiate their task and has given them with sufficient information on, where to start [8].The client is also responsible for gathering and aggregating the results. Savio S.H Tse, proposed three task scheduling algorithms for balancing the load based on two independent criteria with object reallocation [9].

Tse, proposed online Bounds on balancing scheduling algorithm based on two independent criteria with Replication and reallocation. This algorithm balances the loads and storage spaces among homogeneous servers [10]. This algorithm considers only loads and storage spaces and does not consider CPU time and memory requirements.

A. Min – Min Algorithm

The Min-Min task scheduling algorithm locates the task which has lowest execution time. It finds the resource which has minimum completion time and then allocates the task to that resource. The ready time of resource is updated. This procedure is continually executed until all tasks are scheduled.

B. Minimum Completion Time

Minimum Completion Time Grid task scheduling algorithm locates the resource which has completion time is minimal and then allocates the task to that resource. Task scheduling can be carried out based on completion time of resource. Completion time is computed by adding together the execution time and the resource ready time.

C. Minimum Execution Time

Minimum Execution Time scheduling algorithm locates the task which has minimum execution time. The task scheduling can be carried out based on first come first served basis. This algorithm does not consider the resource availability and its load. The limitation of this algorithm is severe load imbalance.

D. Balanced Minimum Execution Time

MET algorithm locates the task which has minimum execution time. It allocates the task to the machine based on first come first served basis. Rescheduling can be performed based on maximum completion time of the task. Thus it increases the resource utilization and load is balanced. The limitations of this algorithm are finding priority of job which is a tedious one and higher turnaround time [11].

E. Load Balanced Min-Min algorithm

T. Kokilavani et al proposed LBMM scheduling algorithm which generates better results than min-min grid task scheduling algorithm. It reduces the makespan and balances the load. The response time is improved. First min-min scheduling algorithm is executed first and rescheduling takes place based on maximum execution time.

F. Highest Response Next Scheduling (HRN)

The jobs are assigned to the resources based on priority and processor competence. Task Scheduling can be performed

International Journal of Communication and Networking System
Volume: 05 Issue: 02 December 2016,Pages No.96-100
ISSN: 2278-2427

based on priority, time, memory and CPU requirements. HRN scheduling algorithm makes use of the resources effectively than First Come First Served scheduling algorithm.

G. Min-Mean heuristic algorithm

Kamalam et al [8] have proposed Min-Mean heuristic grid scheduling algorithm to attain better performance. This algorithm reschedules the Min-Min schedule based on the makespan of all the resources. This algorithm minimizes the makespan than the Min-Min algorithm in a heterogeneous environment.

III. PROBLEM STATEMENT

Task scheduling problem is NP-Complete. Consider T1, T2, T3, T4 and T5 are independent tasks. Each meta-task is scheduled to the resource based on the order in which the various tasks are arrived. The tasks that have no dependency with each other and no priority specified to them.

In the heterogeneous environment the size of the meta-task and number of resources are known. The estimate of the execution time for every task on each resource is known prior to execution. Expected Time to Compute Matrix ETC(Ti, Rj) contains the execution time of each task on each resource. Where Ti represents meta-task and Ri represents Resource Set. This is the precondition of the tasks. The post condition shows that the sequence in which the tasks are executed in order to minimize the makespan. The Problem can be defined as given below:

Consider the task set Ti= T1, T2, T3, T4… Tn and the Resource Set Ri = R1, R2, R3, R4,…….,Rn. The intention of the tasks scheduling algorithm is to make use of the idle time of the resources effectively and reduces the makespan by rescheduling the resources. In order to reduce the makespan, the load is distributed among the available resources. The makespan can be computed as follows:

Makespan = max (CT (Ti, RTj))
CTij = RTj + ETij
CT = Completion Time of the task
RTj = Ready Time of the Resource j.
ETij = Execution time of the Task i on Resource j.

Grid task scheduling problem is used to give the solution with fewer costs.

IV. COMPARATIVE ANALYSIS OF GRID TASK SCHEDULING ALGORITHM

Heuristic algorithm is generally used for allocation and resource reservation. Some of the heuristics tasks scheduling algorithms are Min-min, Load balanced Min-Min, Minimum Execution Time and Balanced Minimum Execution Time [12]. Let us consider the problem comprises two resources namely R1 and R2 and the task group Ti consists of four tasks T1, T2, T3 ad T4. The scheduler schedules the tasks to the available resource R1 and R2. Load imbalance is the main issue in Min-Min algorithm. Balanced Minimum Execution Time scheduling algorithm reduces the makespan compared to Minimum Execution Time Algorithm. Table 1 shows the execution time of the task.

Table 1: Execution Time of the Task in Milliseconds

Task	Resource	
	R1	**R2**
T1	21	16
T2	31	7
T3	22	16
T4	8	9

A. Min – Min

Min-Min algorithm locates the task which has lowest execution time. It allocates the task to the resource which produces minimum execution time. The ready time of resource is updated. The repetitive processing would be required until all tasks are scheduled. The Min–Min algorithm selects the smaller task first and then chooses the larger task. There are some drawbacks in this algorithm if the smaller number of tasks exceeds the larger one.

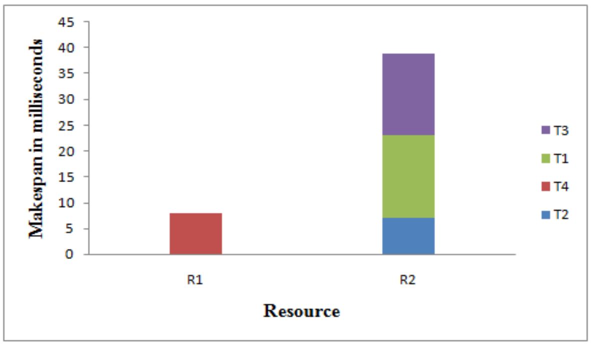

Figure 1: Min-Min Task Scheduling

The Min–Min, Max–Min task scheduling algorithms schedule the task based on the execution time, completion time and availability of resource [7]. Min–min scheduling prefers the minimum execution time. In this example all the tasks T1, T2, T3 are assigned to Resource R2 and the task T4 is assigned to resource R1. In this illustration makespan of Min-Min algorithm is 39 sec. Figure 1 shows the performance of Min-Min grid task scheduling algorithm. Load imbalance problem is the main issue in min-min scheduling algorithm. To avoid these load imbalance problems many heuristic algorithms have been proposed. One of the algorithms is Load balanced Min-Min scheduling algorithm [7].

B. Load Balanced Min-Min (LBMM)

This algorithm finds the resource which has Minimum Execution Time for the particular task. First min-min scheduling algorithm is executed and rescheduling takes place based on maximum completion time. Completion time is computed by adding together the execution time and the resource ready time. It produces better results than Min-Min scheduling algorithm.

Completion Time = Execution Time + Resource Ready Time

In this example the task T1 and T2 scheduled in Resource R2 and the task T3 and T4 are assigned in Resource R1. The Makespan of this algorithm is 30 sec. This algorithm reduces the makespan and balances the load.

International Journal of Communication and Networking System
Volume: 05 Issue: 02 December 2016,Pages No.96-100
ISSN: 2278-2427

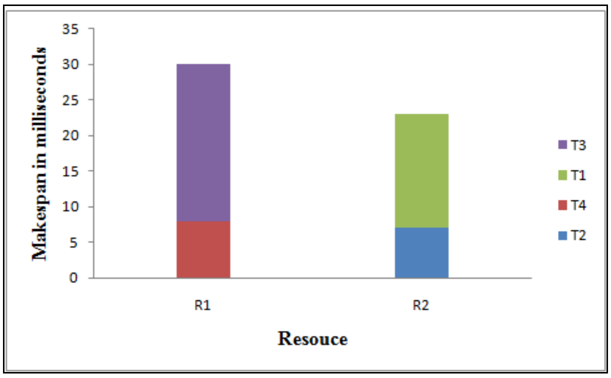

Figure 2: Load Balanced Min-Min Task Scheduling

C. Minimum Execution Time (MET)

This algorithm locates the task which has minimum execution time and allocates the task to the resource based on first come first served basis. Severe load imbalance is the major issue in this algorithm. It does not consider the resource availability and its load. In this example the tasks T1, T2, T3 scheduled in Resource R2 and the task T4 assigned to Resource R1. The makespan of this algorithm is 39. Figure3 shows the performance of Minimum Execution Time Task Scheduling algorithm.

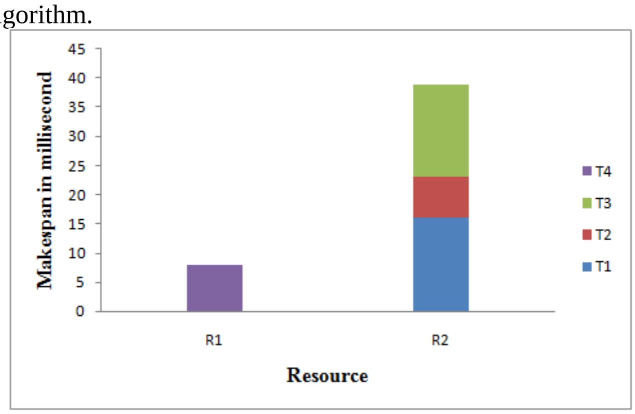

Figure 3: Minimum Execution Time Task Scheduling

D. Balanced Minimum Execution Time (BMET)

M. Hemamalini and Dr. M.V. Srinath proposed Balanced Minimum Execution Time Algorithm. This algorithm maximizes the resource utilization and balances the load. MET algorithm locates the task which has lowest execution time and allocates the task to the resource based on first come first served basis. Rescheduling can be performed based on maximum completion time of the task. Thus it increases the resource utilization and load is balanced. The limitations of this algorithm are finding priority of job is tedious one [11]. This algorithm balances the load and increases the resource utilization. Figure 4 shows the Performance of Balanced Minimum Execution Time algorithm in a heterogeneous environment. In this example the task T1 and T2 scheduled in Resource R2 and the task T3 and T4 are assigned in Resource R1. The Makespan of this algorithm is 30 sec. Hence the load balancing is achieved and resource utilization is increased compared to MET grid task scheduling algorithm.

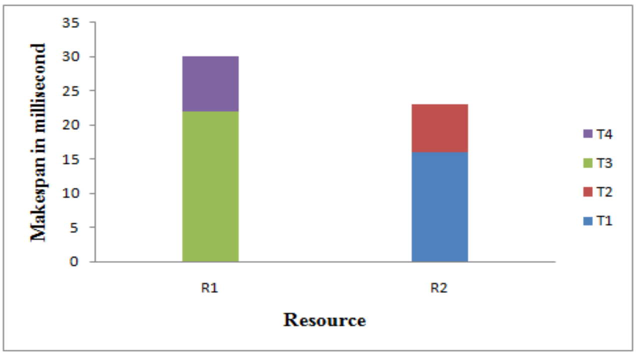

Figure 4: Balanced Minimum Execution Time Task Scheduling

V. EXPERIMENTS AND RESULTS

To assess the performance of scheduling algorithms, consider the problems with both resource and task heterogeneity are collected [5], [8], [16], [17] and implemented for four grid task scheduling algorithms. Figure 5 shows the static mapping of tasks to machine for four scheduling algorithms. Let us consider the problem set P comprises the both resource heterogeneity and task heterogeneity to analyze the performance of BMET scheduling algorithms. P1, P2, P3, P4 and P5 represent five different Problem set. Table2 shows the results obtained for diverse scheduling algorithms. The four different algorithms were compared

Table 2: Problem Set With Results (Makespan in Milliseconds)

Problem Set	Min-Min	LBMM	MET	BMET
P1	58	51	58	51
P2	39	30	39	30
P3	52	43	52	41
P4	60	49	60	48
P5	56	46	56	45

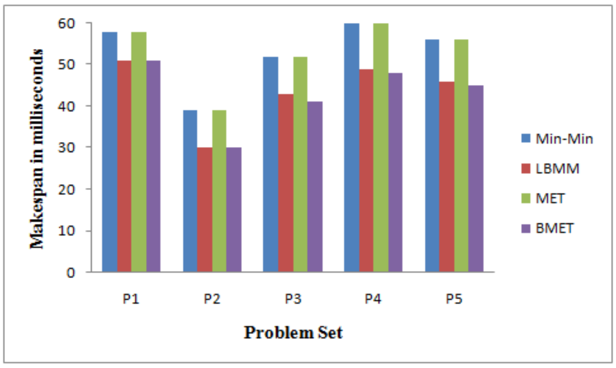

Figure 5: Performance analysis of BMET with other grid task scheduling algorithms

Figure 5 shows how the Balanced Minimum Execution Grid task scheduling algorithm better than that of other scheduling algorithms. From this Gantt chart we can examine that BMET produces less makespan than Min-Min, LBMM and MET scheduling algorithm.

A. Performance analysis of Balanced Minimum Execution Time with other scheduling Algorithms

Table 3: Makespan Produced By Four Grid Task Scheduling Algorithms

Problem set	Resource	Min-Min	LBMM	MET	BMET
P1	R1	58	34	58	34
	R2	16	51	16	51
P2	R1	8	30	8	30
	R2	39	23	39	23
P3	R1	22	33	22	22
	R2	13	19	13	26
	R3	52	43	52	19
	R4	18	27	18	41
P4	R1	26	39	26	26
	R2	17	23	18	32
	R3	60	49	60	24
	R4	22	31	22	48
P3	R1	24	36	24	24
	R2	15	21	15	29
	R3	56	46	56	21
	R4	20	29	20	45

Table 3 shows Makespan produced by four grid Task scheduling Algorithm. This table shows how BMET strives to use the resources effectively than Min-Min, LBMM and MET. Gantt chart shows the Resource utilization. From this chart we can study that Balanced Minimum Execution Time algorithm effectively utilize the resources and reduces the makespan than other algorithms. Resource utilization is calculated using the formula [7].

$$UR = Ti * 100 / TQRU$$

$$TQRU = \sum_{I=1}^{n} CT$$

TQRU = Total Quantity of Resource Used
Ti = Meta task
UR = Usage of Resource
CT = Completion Time of Task

Min–Min scheduling prefers the minimum completion time. In this example Problem set P1, the tasks T1 ,T2 and T4 scheduled in Resource R1 and the task T3 are assigned in Resource R2. The Makespan of this algorithm is 58 ms. MET scheduling prefers the minimum execution time. The tasks T1, T2 and T4 scheduled in Resource R1 and the task T3 are assigned in Resource R2. The Makespan of this algorithm is 58 ms.The resource utilization for R1 is 100 % and 27.5% in R2. In balanced Minimum Execution Time grid task scheduling algorithm, the resource utilization for R1 is 66.6 % and 100% in R2. The experimental results achieved by comparing four different scheduling algorithms for various problem set shows that BMET grid task scheduling algorithm minimizes the makespan than Min-Min, LBMM and MET scheduling algorithms. BMET scheduling algorithm increases the resource utilization and the load is balanced. Table 4 shows the resource utilization of four scheduling algorithms (in percentage). Balanced Minimum Execution Time algorithm utilizes the resource effectively than Min-Min, LBMM and MET. Figure 6 shows the resource utilization in percentage. This experimental outcome shows that BMET grid task scheduling algorithm produces better results than the other scheduling algorithms.

Table 4: Resource Utilization of Four Grid Task Scheduling Algorithms in Percentage

Problem set	Resource	Min-Min	LBMM	MET	BMET
P1	R1	100	66.6	100	66.6
	R2	27.5	100	27.5	100
P2	R1	20.5	100	20.5	100
	R2	100	76.6	100	76.6
P3	R1	42.3	76.7	42.3	53.6
	R2	25	44.1	25	63.4
	R3	100	100	100	46.3
	R4	34.6	62.7	34.6	100
P4	R1	43.3	79.5	43.3	54.1
	R2	28.3	46.9	28.3	66.6
	R3	100	100	100	50
	R4	36.6	63.2	36.6	100
P3	R1	42.8	78.2	42.8	53.33
	R2	26.7	45.6	26.7	51.7
	R3	100	100	100	46.6
	R4	35.7	63	35.7	100

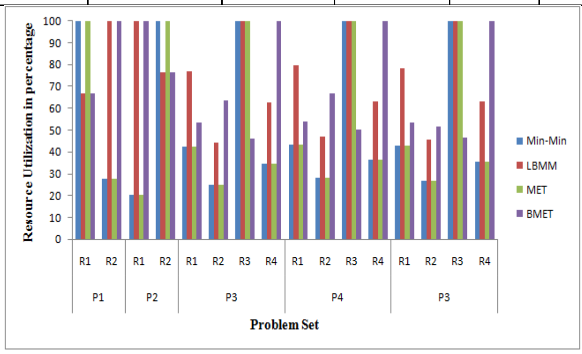

Figure 6: Resource utilization of four Grid Task scheduling Algorithms in Percentage

VI. CONCLUSION

Task scheduling is an important area in the Grid computing scenario. Efficient Grid task scheduling algorithm is essential to make use of the resource effectively and minimizes the overall completion time of resources. The main purpose of grid task scheduling algorithm is to maximize the throughput based on availability of resources. In this paper we analyzes the performance of Balanced Minimum Execution Time algorithms with other algorithms such as Min-Min, LBMM, Minimum Execution Time based on execution time, makespan, completion time and load balancing. This survey shows that Balanced Minimum Execution Time outperforms the other grid task scheduling algorithms. BMET grid task scheduling algorithm utilizes the resource effectively and produces better result than other algorithms. Future directions of this study can be extended by suggesting a new hybrid algorithm which combines the advantages of all the four algorithms. The Quality of service characteristics such as scheduling cost and memory requirement of the task can be considered as a future scope.

References

[1] Lei, Zhou and Yun, Zhifeng, Allen, Gabrielle., Grid Resource Allocation. [ed.] Lizhe Wang, Wei Jie and Jinjun Chen. *Grid Computing: Infraestructure, Service, and Applications.* Boca Raton : CRC Press, 7, pp. 1172-188, 2009.

[2] M.Hemamalini, Dr.M.V.Srinath, State of the Art: scheduling Algorithms in a Heterogeneous Grid computing Environment, Elysium Journal Engineering and research and Management,August 2014, volume 1, Issue 1.

[3] HE Xiaoshan, Xian –He sun, Gregor von Laszewski. Qos Guided Min-Min Heuristic for Task Scheduling. International Journal of Computer Applications, 2003.

[4] Saeed Parsa, Reza Entezari- Maleki. RASA: A New Scheduling Algorithm in Grid environment. International Journal of Digital Content Technology and its Applications. Volume 3, Number 4, 2009.

[5] Sameer Singh Chauhan, R.C.Joshi. Qos guided Algorithms for Task Scheduling. International journal of computer applications, Volume 2. No.9, 2010.

[6] Dr.K.Vivekanandan, D.Ramyachitra. A Study on Task Scheduling in Grid environment. International Journal on Computer Science and Engineering, ISSN: 0975-3397 Vol. 3 No. 2, 2011.

[7] Dong. F, Luo. J, Gao. L and Ge. L. A Task Scheduling Algorithm Based on QoS Priority. In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC’06), IEEE, 2006.

[8] Nadar Mohammed and Jameela Al-Jaroodi, Delay-Tolerant Dynamic Load Balancing,IEEE International Conference on High performance Computing and Communications,2011.

[9] Kassian Plankensteiner, Radu Prodan, Thomas Fahringer, A New Fault Tolerance Heuristic for ScientificWorkflows in Highly Distributed Environments based on Resubmission Impact, IEEE International Conference on e-Science,2009.

[10] Savio S.H.Tse , Online Bounds On Balancing Two Independent Criteria With Replication and Reallocation, In IEEE Transactions on Computers”,2011.

[11] M.Hemamalini, Dr.M.V.Srinath, Balanced Minimum Execution Time Algorithm in a distributed heterogeneous environment, Proceedings of IEEE International Conference on Advances in Engineering and Technology, 2014.

[12] M.Hemamalini , Review on Task scheduling Algorithm in Heterogeneous Distributed Environment, International Journal of computer Applications, vol. 40 , No.2, ISSN:0975-8887, p24-30, 2012.