

Review on Grid Task Scheduling in Distributed Heterogeneous Environment

M. Hemamalini
Assistant Professor (Sl.Grade),
Department of Computer Applications,
A.V.C College of Engineering, Mayiladuthurai.
Tamil Nadu, India.

ABSTRACT

Computational grid is a potential technology mainly used for distributed environment. The major issues related with Grid are resource discovery, heterogeneity, fault tolerance and task scheduling. Grid task scheduling is an integrated component of computing which effectively utilizes the idle time of resources. Efficient scheduling algorithm is needed to utilize the resources effectively and reduce the overall completion time. This paper analyzes the performance of scheduling algorithms from different point of view such as makespan, execution time, completion time and load balancing. First, the general view of World Wide Web grid computing environment and its functions are discussed. Then this paper examines the performance of four scheduling algorithms such as Min-Min, Max-Min, Minimum Completion Time and Minimum Execution Time. Based on the comparative study of various algorithms, some common issues are proposed. The conventional Max-Min grid task scheduling algorithm effectively utilizes the resources and minimizes the makespan than other scheduling algorithms. This survey shows that Max-Min grid task scheduling algorithm outperforms the other algorithms in both task and resource heterogeneous environment.

Keywords: Grid Computing, Task scheduling, Heterogeneous computing, Makespan.

1. INTRODUCTION

Grid computing emphasizes coordinating and sharing of computational power, data storage and resources in the dynamic and geographically distributed environment. The main goal of scheduling is to increase the throughput and efficient utilization of the resources. Grid is defined by Ian Foster [1] as flexible, secure, large scale resource sharing among dynamic collection of individuals, institutions and resources which is called as virtual organizations.

Grid architecture has been proposed based on the creation of Virtual Organizations. In a World-Wide grid environment, the infrastructure needs various capabilities such as supporting remote storage, uniform access to remote resource, security, discovery of computational resource, mapping and scheduling of tasks [2].

1.1 Grid Resource Broker

Grid Resource Broker is responsible for resource discovery, allocation of a job to a particular resource and binding of user applications. When the client submitted the job to the Grid Resource Broker, the broker splits the job into various tasks and distributes them to several resources according to user's requirements and availability of resources [2].

1.2 Grid Information Services

Grid resources are registered in Grid Information Services. Grid Information Service is used to keep the information about all resources. The user specifies their service requirements and submits to resource broker. Grid Resource Broker discovers the resource characteristics using Grid Information Service. Resource Broker schedules the task for execution on these available resources. Grid environment entails the services like security, resource allocation, execution management and scheduling [2]. Figure 1 illustrates general view of World-Wide Web grid computing environment.

1.3 Heterogeneous Computing

Heterogeneous Computing refers the interconnection between different high performance machines and high-speed links in a distributed environment. It is used to meet the computational demands of large and divergent task [3].

1.4 Grid Task Scheduling

One of the essential Components of parallel and distributed computing is Task Scheduling. With emergence of grid computing, new scheduling algorithm is in demand for Grid environment. Parallel scheduling algorithm schedules the task in order to reduce turn around time. The main drawback of parallel scheduling is ignoring the specific shared nature of the resource. In a computational grid, scheduling problem is enhanced by maximizing system utilization and fulfills economical system and user constraints. [4]

Grid task scheduling is an integrated component of computing deals with static or dynamic policies, application models and quality of service constraints like bandwidth, objective functions and dynamic behavior of resources.

Several task scheduling algorithms are existing to minimize the makespan [5], [10], [13], [14], [15]. With the explosion of the Grid, there are two conception need to be considered in a scheduling model such as non-dedicated network and Quality of Service.

In a non-dedicated network, users have their own local jobs and cannot provide exclusive services to remote jobs. We need to predict the computation time for non-dedicated networks [5]. In a computational grid, it is desirable to compete for the best QoS for remote resources to fulfill application constraints [5].

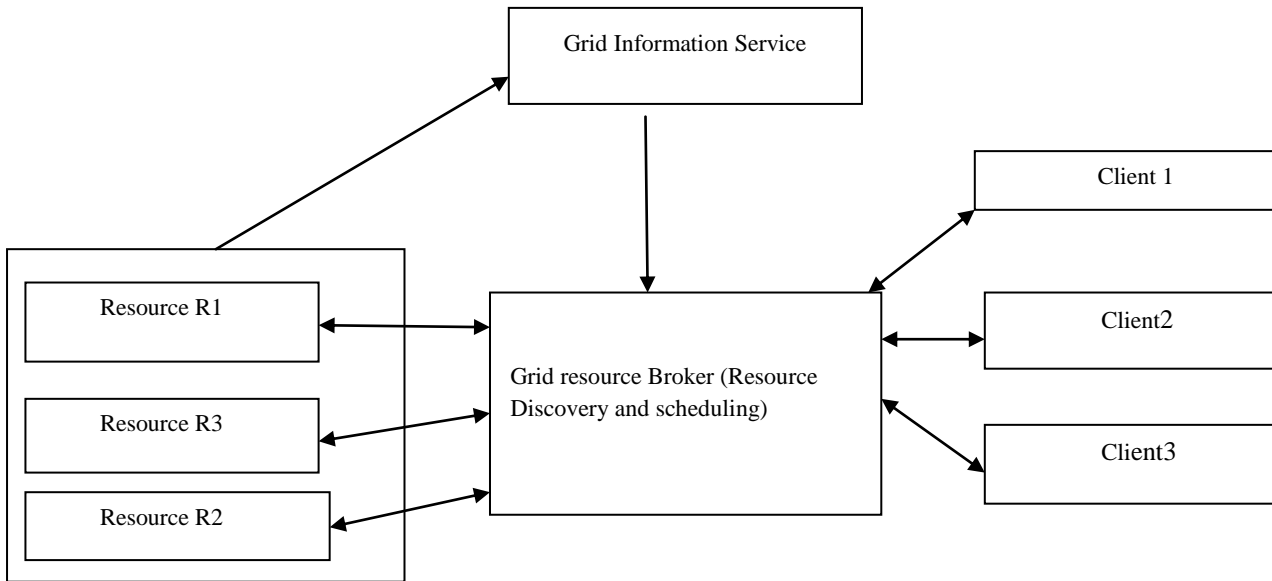


Figure 1 – General View of World –Wide Web Grid computing environment.

He Xiaoshan, Xian-He Sun have stated that the scheduler in the Grid environment needs to consider application and QoS constraints to get a better match between applications and resources. Prediction model can be adopted to address the challenges of non-dedicated network.

2. REVIEW OF LITERATURE ON SCHEDULING ALGORITHM

2.1 Min – Min Algorithm

This algorithm finds the task which has minimum execution time and assigns the task to the resource that produces minimum completion time. The ready time of resource is updated. This procedure is repeatedly executed until all tasks are scheduled.

2.2 Max-Min scheduling algorithm

The Max-Min scheduling algorithm is similar to Min-Min scheduling algorithm but it schedules the larger task first. The ready time of resource is updated. This process is repeated until all unmapped tasks are assigned.

2.3 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.

2.4 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.

2.5 Highest Response Next Scheduling (HRN)

Here the jobs are allocated to processor based on priority and processor competence. Scheduling is based on priority, time, memory and CPU requirements. HRN effectively utilize the resources than First Come First Served scheduling algorithm.

The main drawbacks of this algorithm are finding priority of job is tedious one and higher turnaround time [6].

2.6 Optimal Resource Constraint Scheduling

It distributes the task among the available processor based on processor capability. It is an efficient load balanced task scheduling algorithms which reduces the turnaround time and average waiting time. It is suitable for more number of jobs and avoids starvation problem.

T.Kokilavani et al proposed LBMM scheduling algorithm which produces better results than min-min scheduling algorithm. It reduces the makespan and tries to balance the load. The response time is improved and load balancing is achieved efficiently. This algorithm applies the min-min scheduling in the first phase and rescheduling takes place based on maximum execution time. Resource utilization for a particular problem is calculated using the formula [7].

$$RU = Ti * 100 / TARU$$

$$TARU = \sum_{i=1}^n CT$$

TARU = Total amount of Resource used.

CT = Completion Time

RU = Resource Used

Kalam et al [8] have proposed a scheduling algorithm called Min-Mean heuristic scheduling algorithm to achieve better performance. This algorithm reschedules the Min-Min schedule based on the mean makespan of all the resources. This algorithm reduces the makespan than the Min-Min algorithm in a heterogeneous environment.

A Job schedule Model based on grid environment (JSMB) consists of three nodes such as Supervisor node, Supervisor Backup node, and Execute Grid node [9]. If the supervisor node is failed then the Backup node will be an active node and balance the load. It reduces the turnaround time and increases the CPU Utilization.

Saeed Parsa et al proposed RASA grid task scheduling algorithm. First the numbers of available resources are calculated. If the available resource is even, the Max-min grid task scheduling algorithm is applied to schedule the first task, otherwise the min-min scheduling algorithm is applied. For example, if the first task is scheduled by Max-min algorithm, the next task will be scheduled by Min-min scheduling. This algorithm ignores the waiting time of the large task in min-min scheduling [10].

HE Xiaoshan et al proposed QoS Guided Min-Min heuristic algorithm. QoS requirement is one of the important factor in grid task scheduling. It is similar to Min-Min scheduling algorithm. Scheduling can be performed based on QoS parameter such as network bandwidth. The two task groups are High QoS and Low QoS. Normally the task requiring high QoS can run on high QoS providing host. It maps the task with high QoS request first. For each task with high QoS request, the algorithm finds the earliest completion time and the host that obtains it, in all the QoS qualified host sets [11].

The algorithm finds the tasks with the minimum earliest completion time and assigns the task to the host that gives the earliest completion time to it. The algorithm finalizes by deleting the scheduled task from the meta-task. The overall performance of the grid system is improved. The QoS guided Min-Min heuristic first schedules the task from high QoS group on resources that can provide high QoS as required. And then it schedules task from low QoS group [5].

The term Quality of Service is used differently based on its perspective. In a network environment QoS represents the bandwidth for the applications. QoS for processor represents the speed or processor utilization.

Let the grid composed of non – dedicated host and each host is composed of several computational resources, which may be homogeneous or heterogeneous. The performance of computing hosts is measured by these parameters such as utilization, arrival rate, standard deviation of service time, network bandwidth and latency.

Grid scheduler does not have control over them without the exclusive control of the local host. The lack of control is the main problem. Grid scheduler must take effort decision and then submit the job to the host by selecting the best match from the pool of applications and the available hosts.

3. PROBLEM STATEMENT

Task scheduling is one of the NP-Complete problems. Let T1, T2, T3, T4 and T5 are collection of indivisible and independent tasks. The tasks that have no dependency among each others are referred as meta task. Each task is assigned to the resource based on the order in which the tasks are arrived. The mapper controls the execution of all tasks and the tasks have no priority and deadline specified them.

In the heterogeneous environment the size of the meta task and number of resources are known. The estimate of the expected execution time for each task on each resource is known prior to execution. Expected Time to Compute Matrix ETC (Ti, Rj) contains the execution of each task. Where Ti represents meta-task and Rj represents Resource Set. The Problem can be defined as follows:

Let task set $T_i = T_1, T_2, T_3, T_4, \dots, T_n$.

Let Resource Set $R_i = R_1, R_2, R_3, R_4, \dots, R_n$.

The main aim of the scheduling algorithm is to effectively utilize the idle time of the resources and minimizes the makespan. The makespan can be calculated as follows:

$$\begin{aligned} \text{Makespan} &= \max (CT (T_i, R_j)) \\ CT_{ij} &= R_j + ET_{ij} \\ CT &= \text{Completion Time} \\ R_j &= \text{Ready Time of Resource } j. \\ ET_{ij} &= \text{Execution time of Task } i \text{ on Resource } j. \end{aligned}$$

Grid task scheduling is one of the NP-Complete Problem used to find the acceptable solution with fewer cost.

4. COMPARATIVE ANALYSIS OF GRID TASK SCHEDULING ALGORITHM

Heuristic algorithm is widely used for allocation and resource reservation. Some of the heuristics scheduling algorithms are Min-min, Max-min, Minimum Completion Time and Minimum Execution Time [12].

Let us consider the problem having two resources R1 and R2 and the task group Ti consists of four tasks T1, T2, T3 ad T4. The scheduler schedules all the tasks based on the available resource R1 and R2. One of limitation of Min-Min scheduling algorithm is load imbalance.

Minimum Completion Time minimizes the makespan compared to Minimum Execution Time Algorithm. Max-Min scheduling algorithm produces better results than other scheduling algorithm in heterogeneous environment. Table 1 shows the execution time of the task.

Table 1: Execution Time of the Task

Task/ Resource	R1	R2
T1	150	18
T2	32	7
T3	20	3
T4	50	15

4.1 Min – Min Algorithm

Min-Min algorithm finds the task which has minimum execution time and assigns the task to the resource that produces minimum completion time. The ready time of resource is updated. This procedure is repeatedly executed until all tasks are scheduled. The Min-Min scheduling algorithm chooses the smaller task first. There are some drawbacks if the smaller number of tasks exceeds the larger one.

The Min-Min, Max-Min task scheduling algorithms work by considering the execution time, completion time and availability of resource [7]. Min-min scheduling prefers the minimum completion time. In this example all the tasks are assigned to Resource R2 and R1 becomes idle. In this illustration makespan of Min-Min scheduling algorithm is 43 sec. Figure 2 shows the performance of Min-Min Grid Task Scheduling algorithm.

Main drawback of this algorithm is unbalanced load. To avoid these problems many algorithm have been proposed. One of the algorithms is Load Balanced Min-Min scheduling algorithm [7].

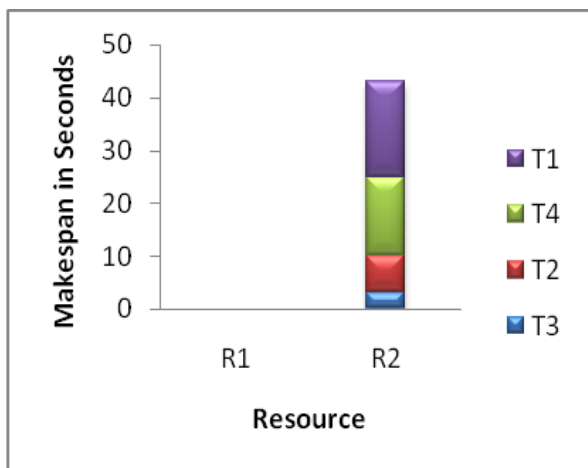


Figure 2: Performance of Min-Min Task Scheduling

4.2 Max-Min scheduling algorithm:

The Max-min scheduling algorithm is similar to Min-min scheduling algorithm but it schedules the larger task first. The ready time of resource is updated. This process is repeated until all unmapped tasks are assigned. Min-min and Max-min are used for small scale distributed system. In this illustration the tasks T1, T4 and T3 are scheduled in Resource R2 and the task T2 assigned in Resource R1. It reduces the waiting of the larger task and produces the load balanced schedule. A Max-min algorithm makes use of the idle resource R1 and reduces the makespan to 36 sec which is lesser than other scheduling algorithm. So Max-min produces the better result. Figure 3 illustrates the performance of Max-Min Grid Task Scheduling algorithm.

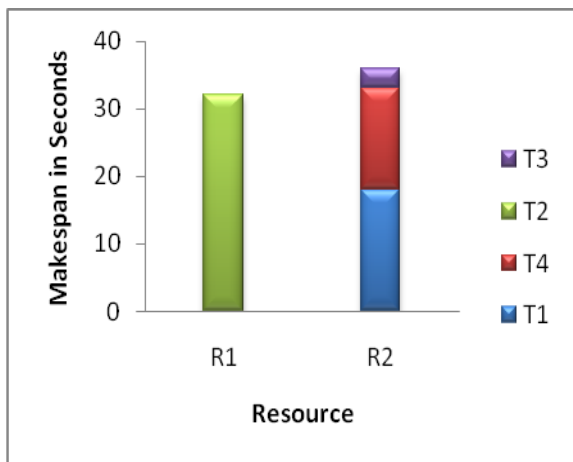


Figure 3: Performance of Max-min Task Scheduling

4.3 Minimum Completion Time (MCT):

This 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 and the ready time of the resource.

$$\text{Completion Time} = \text{Execution Time} + \text{Resource Ready Time}$$

In this example the task T1, T2 and T4 scheduled in Resource R2 and the task T3 assigned in Resource R1. The Makespan of this algorithm is 40 sec.

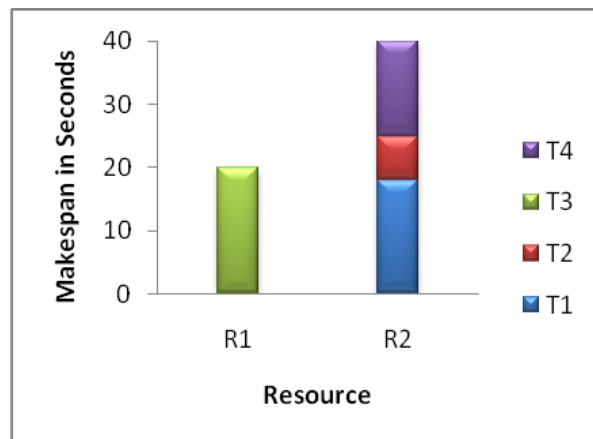


Figure 4: Performance of Minimum Completion time Task Scheduling

4.4 Minimum Execution Time (MET):

This algorithm finds the task which has minimum execution time and assigns the task to the resource based on first come first served basis. One of the main drawbacks of this algorithm is load imbalance.

It does not consider the availability of the resource and its load. In this example the tasks T1, T2, T3 and T4 scheduled in Resource R2 and the Resource R1 becomes idle. The makespan of this algorithm is 43. Figure 5 shows the performance of Minimum Execution Time Task Scheduling algorithm.

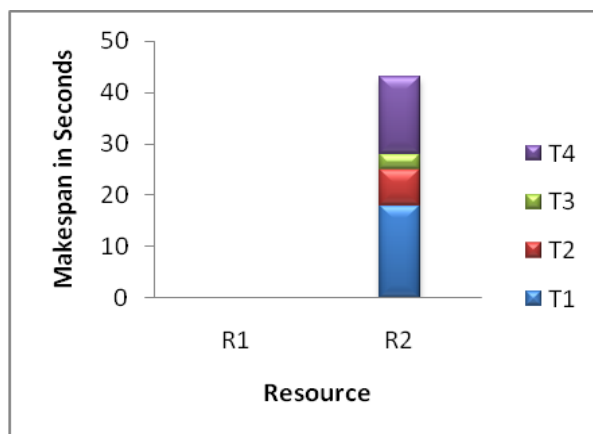


Figure 5: Performance of Minimum Execution Time Task Scheduling

5. EXPERIMENTS AND RESULTS

To assess the performance of scheduling algorithms, problems having both resource and task heterogeneity are composed from various literature [5], [8], [16], [17] and implemented for four grid task scheduling algorithms. Figure6 shows the static mapping of tasks to machine for four scheduling algorithm. To analyze the four scheduling algorithms, let in problem set P comprises the both resource heterogeneity and task heterogeneity.

P1, P2, P3, P4 and P5 represent five different Problem set. Table2 shows the results obtained for diverse scheduling algorithms.

Table 2: Problem set with results (Makespan in seconds)

Problem Set	Min-min	Max-min	MCT	MET
P1	43	36	40	43
P2	84	50	69	84
P3	314	221	221	314
P4	362	225	275	362
P5	281	201	281	281

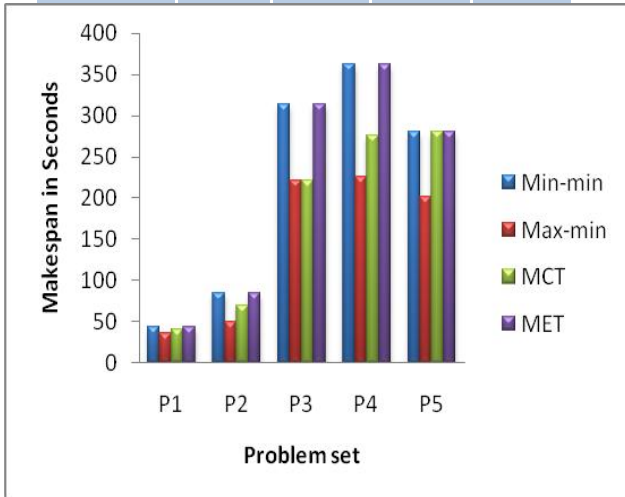


Figure6: Comparative analysis of four grid task scheduling Algorithm

Figure6 shows how the Max-Min algorithm outperforms the other scheduling algorithm. From this Gantt chart we can examine that Max-Min produces less makespan than other scheduling algorithm.

5.1 Resource Utilization for four scheduling Algorithm:

Table 3: Resource Utilization (Makespan in Seconds)

PROBL EM SET	MIN-MIN		MAX-MIN		MCT		MET	
	R1	R2	R1	R2	R1	R2	R1	R2
P1	0	43	32	36	20	40	0	43
P2	4	84	43	50	69	32	4	84
P3	314	127	221	222	221	222	314	127
P4	362	0	225	160	122	275	362	0
P5	281	31	201	121	281	31	281	31

Table 3 shows the resource utilization of four diverse scheduling algorithms. This table shows how max-min strives to use the resources effectively. Gantt chart shows the

resource utilization. From this chart we can study that max-min scheduling algorithm uses the maximum amount of resources and reduces the makespan. Resource utilization for a particular problem 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, all the tasks are assigned to Resource R2 and R1 becomes idle. So resource utilization for R1 becomes zero percentage and 100 percentages for R2.

In Max-min scheduling algorithm 100% utilization of resource R2 and 88.8 % utilization of resource R1 for the problem set P1. In this illustration Makespan of Max-min scheduling algorithm is lesser compared to other scheduling algorithm.

MET scheduling prefers the minimum execution time. All the tasks are assigned to resource R2 and R1 is idle. The resource utilization for R2 is 100 % and 0% in R1. In Minimum Completion Time scheduling algorithm, the resource Utilization for R2 is 100% and R1 is 50%.

The experimental results achieved by comparing four different scheduling algorithms for various problem set shows that Max-min scheduling algorithm minimizes the makespan than other scheduling algorithms. Table 4 shows the resource utilization of four scheduling algorithms (in percentage).

Table 4: Resource Utilization in Percentage

Problem Set	Algorithm Used	R1	R2
P1	Min-min	0	100
	Max-min	88.8	100
	MCT	50	100
	MET	0	100
P2	Min-min	4	100
	Max-min	86	100
	MCT	100	46.3
	MET	4	100
P3	Min-min	100	40.4
	Max-min	99.5	100
	MCT	99.5	100
	MET	100	40.4
P4	Min-min	100	0
	Max-min	100	71.1
	MCT	44.3	100
	MET	100	0
P5	Min-min	11	100
	Max-min	60	100
	MCT	11	100
	MET	11	100

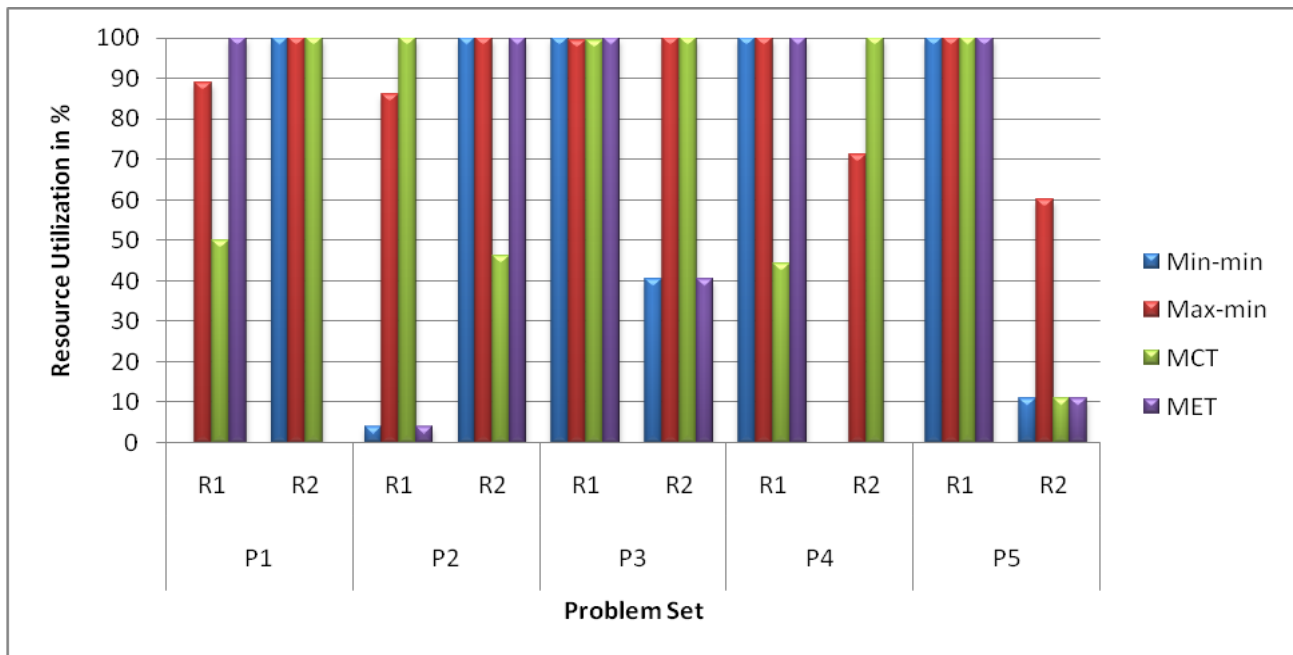


Figure 7: Resource utilization in Percentage

Max-Min grid task scheduling algorithm utilizes the resource effectively and produces better result. Figure7 shows the resource utilization in percentage. This experimental outcome shows that Max-Min grid task scheduling algorithm outperforms the other algorithm in a heterogeneous distributed environment.

6. CONCLUSION

Task scheduling is an important issue in the recent Grid computing scenario. Efficient task scheduling algorithm is needed to utilize the resource effectively and reduce the overall completion time. The main goal of grid task scheduling is to increase the throughput based on availability of resources. In this paper four different scheduling algorithms are compared such as Min-min, Max-min, MCT and MET. The experimental results achieved by comparing four different scheduling algorithms for various problem set shows that Max-min scheduling algorithm minimizes the makespan than other scheduling algorithms. Max-min grid task scheduling algorithm utilizes the resource effectively and produces better result. This paper identifies that Max-min grid task scheduling algorithm outperforms the other algorithm in a heterogeneous distributed environment. This study can be further extended by proposing a new hybrid algorithm which combines the advantages of all the four algorithms.

7. REFERENCES

- [1] Ian Foster, Carl Kesselman, Steven Tuecke. 2001. The Anatomy of the Grid Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*.
- [2] Rajkumar Buyya and Sri kumar Venugopal. 2005. A Gentle Introduction to Grid Computing and Technologies Computer Society of India.
- [3] Braun, T.D., Siegel, H.J., Beck, N., Boloni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., et al. 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, Vol. 61, No. 6, pp.810–837.
- [4] Buyya, R., Abramson, D. and Giddy, J. Nimrod/G: architecture of a resource Management and scheduling system in a global computational Grid. 2000. *High Performance Computing Asia 2000*, Beijing, China, May 14-17, pp.283-289.
- [5] HE Xiaoshan, Xian –He sun, Gregor von Laszewski. Qos Guided Min-Min Heuristic for Grid Task Scheduling. *International Journal of Computer Applications*.
- [6] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan. 2010. A Survey of Job Scheduling and Resource Management in Grid Computing, *World Academy of Science, Engineering and Technology* 64.
- [7] T.Kokilavani, Dr.D.I.George Amalarethinam. Load Balanced Min-Min Algorithm for static Meta – Task Scheduling in Grid Computing. 2011. *International Journal of Computer Applications*, Volume 20 – No.2.
- [8] Kamalam.G.K and Muralibhaskaran.V. A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-Tasks On Heterogenous Computing Systems . 2010. *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.1.
- [9] Homer Wu, chong – Yen Lee , Wu – yee chen, Tsang Lee. 2007. A Job Schedule Model based on Grid Environment. *IEEE Proceedings of the First international Conference on Complex, Intelligent and software Intensive System*, CISIS'07 2007.
- [10] Saeed Parsa, Reza Entezari- Maleki. 2009. RASA: A New Grid Task Scheduling Algorithm. *International Journal of Digital Content Technology and its Applications*. Volume 3, Number 4.

- [11] Sameer Singh Chauhan, R.C.Joshi. Qos Guided Heuristic Algorithms for Grid Task Scheduling. 2010. International journal of computer applications, Volume 2. No.9.
- [12] Dr.K.Vivekanandan, D.Ramyachitra. A Study on Scheduling in Grid Environment. 2011. International Journal on Computer Science and Engineering, ISSN: 0975-3397 Vol. 3 No. 2.
- [13] Dong. F, Luo. J, Gao. L and Ge. L. A Grid Task Scheduling Algorithm Based on QoS Priority Grouping. 2006. In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), IEEE.
- [14] Etminani .K, and Naghibzadeh. M, A. Min-Min Max-Min Selective Algorithm for Grid Task Scheduling. 2007. The Third IEEE/IFIP International Conference on Internet, Uzbekistan.
- [15] Ullah Munir. E, Li. Jianzhang, and Shengfei, Shi. QoS Sufferage Heuristic for Independent Task Scheduling in Grid. 2007. Information Technology Journal, 6 (8): 1166-1170.
- [16] Geoffrey Falzon, Maozhen Li. Enhancing list scheduling heuristics for dependent job scheduling in grid computing environments. 2010. Journal of Supercomputing, Springer.
- [17] Yagoubi. B, and Slimani, Y. Task Load Balancing Strategy for Grid Computing. 2007. Journal of Computer Science, Vol. 3, No. 3, pp. 186-194.