Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 10 (2017) pp. 2911-2925

http://www.ripublication.com

ETS: An Efficient Task Scheduling Algorithm for Grid Computing

S. Vaaheedha Kfatheen* and Dr. A. Marimuthu

Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India.

Associate Professor, P.G & Research Department of C.S, Government Arts College, Coimbatore, TN, India.

Abstract

Grid computing is a form of distributed computing to solve large-scale scientific problems. Scheduling is found to be a rigorous area of research in grid computing. Scheduling is defined to be a decision process that could assign relatively large number of tasks to the available resources in order to execute the tasks in a short span of time and thereof achieving effective and efficient performance. A mark of better performance may be said as effective utilization of resources by balancing the load across them and less makespan. However, the heterogeneity and dynamic nature of grid makes the development of the scheduling algorithms very difficult. Task scheduling is one of the NP-Complete problems and it becomes the focus of many researchers. The main goal of task scheduling is to minimize the makespan and optimal utilization of resources. In this paper, a new task scheduling algorithm named as ETS (An Efficient Task Scheduling Algorithm for Grid Computing) has been proposed. The proposed ETS algorithm is found to manage load among the available resources and result in better schedules than the compared algorithm. The proposed algorithm has been tested using GridSim Simulation toolkit [1] and various sets of outcomes show that the proposed ETS algorithm produces the minimum makespan and the load on the resources is optimally balanced.

Keywords: Grid Computing, Scheduling, NP-Complete Problem, Makespan, Resource Utilization.

1. INTRODUCTION

Grid computing is a high performance computational system which consists of a large number of geographically distributed heterogeneous resources, linked through the Internet. According to Foster and Kesselman [2], Grid computing has both hardware and software infrastructure which offer a cheap, distributable, coordinated, and reliable access to powerful computational capabilities. Scheduling[3] is a very important problem in grid computing, to map the tasks on to the collection of heterogeneous resources available in a massive geographic spread.

There are different stages of grid scheduling. First stage is resource discovery, which provides a list of available resources. Stage two is resource allocation, which involves the selection of possible resources and the mapping of tasks to the resources. This resource allocation stage is an NP-Complete problem [4]. The third stage includes task execution. Many heuristic algorithms have been devised to solve the task scheduling problem. Two important metrics for measuring the efficiency of the task scheduling are makespan and resource utilization. Makespan [5] is the time when grid computing system completes its latest task. An efficient scheduling algorithm is one which minimizes the idle processing time and maximizes the resource utilization. In this paper, a new task scheduling algorithm for static mapping is presented to achieve better performance. The proposed algorithm has been tested using GridSim Simulation toolkit and the simulation results show that the proposed ETS algorithm produces minimal makespan; also load across the resources is optimally balanced.

2. PROBLEM DEFINITION

In this paper, the problem of scheduling M tasks on N resources is focused. The main goal of task scheduling algorithm is to allocate efficiently the task to the appropriate resource and hence the makespan is minimized. The concept of ETC (Expected Time to Compute) matrix introduced by Shoukat et al., is used to formulate the problem [6]. Each row of ETC matrix contains the estimated execution times of a given task on each resource and each column of the ETC Matrix consists of estimated execution time of a given resource for each task.

To solve the problem mathematically,

let T_i be the number of independent tasks, where i=1,2,3…m

and

M_j be the number of resources, where j=1,2,3,…n

and this can be represented as M x N matrix format.

E_{ij} is the Expected Time to Complete a task ‘i’ (T_i) on resource ‘j’ (R_j).

In order to minimize the makespan and maximize the resource utilization, the tasks of a parallel application have to be efficiently scheduled on the available grid resources. Using ETC matrix model, the proposed algorithm is developed under a set of assumptions:

- ETC Matrix of size m x n, where ‘m’ represents the number of tasks and ‘n’ represents the number of resources.
- Tasks have no priorities associated with them.
- Tasks have no deadlines among them.
- Independent tasks have to be allocated on to the available resources.
- Scheduling process can be performed in batch mode fashion.
- The availability of grid resources and the number of tasks to be executed are known in prior.

3. RELATED WORKS

Many heuristic algorithms have been proposed by various researchers for static task scheduling in grid environment to minimize the makespan [7,8,9,10,11,12]. Due to the dynamic nature and heterogeneity of grid, most of the well known scheduling algorithms are not suitable for large scale systems. Resource utilization aims to increase the busy time of resources. In this proposed algorithm, we focus on resources that are not properly utilized. To start with, the algorithm identifies the resources that are highly loaded and tries to share the load to other less loaded resources or idle resources. Then, the algorithm removes the task from heavily loaded resource and allocates to leisurely loaded resources. Hence, parallelism in resource utilization is considerably increased and makespan could be reduced; also load on resources may get balanced.

Doreen D et al., [13] have proposed an efficient Set Pair Analysis (SPA) based task scheduling algorithm named Double Min-Min algorithm which performs scheduling to enhance system performance in Hypercubic P2P Grid (HPGRID). The simulation result shows that the SPA based Double Min-Min scheduling minimizes the makespan by balancing the load and guarantees high system performance.

Kamalam et al.,[14] have presented a new scheduling algorithm named Min-Mean heuristic scheduling algorithm for static mapping to achieve better performance. The proposed algorithm reschedules the Min-Min produced schedule by considering the mean makespan of all the resources. The algorithm deviates in producing a better schedule than the Min-Min algorithm when the task heterogeneity increases.

Sameer Singh et al., [7] have presented two heuristic algorithms namely QoS Guided Weighted Mean Time-Min(QWMTM) and QoS Guided Weighted Mean Time Min-Min Max-Min Selective(QWMTS). Both algorithms are found to be proposed for batch mode independent tasks scheduling. The network bandwidth is taken as QoS parameter.

Sadegh Nejazadeh et al., [15] have presented a new method to schedule the tasks in computational grids. The goal of the scheduler is to minimize the makespan and execution time. This algorithm and various existing algorithms have been tested using the benchmark simulation model for distributed heterogeneous computing systems.

Gaurang Patel et al., [16] have presented a two-phase load balancing policy for the multi-cluster grid environment where, computational resources are dispersed in different administrative domains or clusters which are located in different local area networks. This algorithm is based on Min-Min Strategy and it uses rescheduling parameter to make use unutilized resources in an efficient way.

The Model

A. Assumptions

The problem considered in this paper consists of scheduling a set of m tasks that need to be processed by a set of n resources. The assumptions of the proposed model are described as follows,

1. Processing time of each task is known in advance.
2. All tasks are available for processing at time zero.
3. Each resource can perform only one task at a time.
4. Once the task has begun on a resource, it must not be interrupted.

B. Mathematical Model

The notations and description used in this mathematical model. Assume

Notations	Description
m	Number of Task
n	Number of Resource
C_{ij}	Completion time of assigning task i on Resource j
A_j	Number of units of processing that resource j has available
T	Total Completion Time
S_{ij}	Starting time of task i on resource j
X_{ij}	Boolean Function

X_{ij} = {0, if the i^{th} task is not assigned to the j^{th} resource 1, otherwise}

Minimum completion time can be achieved by

$$\text{Min(T)} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} \qquad (1)$$

subject to m>0, n>0 and $X_{ij}=1$

$$\sum_{j \in N_i} X_{ij} = 1 \qquad (2)$$

$$C_{ij} >= 0; X_{ij} \in \{0,1\} \qquad (3)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} <= A_j \qquad (4)$$

$$S_{ij} + C_{ij} <= X_{ij} . L \qquad (5)$$

With the reference to constraint (1), minimizing the completion time is main objective.

Constraint (2) make sure that task i is assigned to only one resource. In constraint (3), the value of C_{ij} ensures that the completion time of task i on resource j is executed or not. Constraint (4) shows that all the tasks have been allocated. If task i is not assigned to resource j, the constraint (5) set the starting and completion times of it on resource j equal to zero.

4. METHODOLOGY OF ETS ALGORITHM

The methodology of Task scheduling algorithm (ETS) is presented in Figure 1. The algorithm starts by identifying the tasks of minimum execution time and the respective grid resources in order to assign the task with minimum execution time on to the resource to execute first. On considering the ETC values, if the number of the smaller tasks is more than the number of the larger tasks then this schedule may not be suitable to schedule them appropriately and therefore the makespan may result in a relatively large value. Then makespan, Completion Time (CT) of all the resources, Average Resource Utilization (ARU), Mean square deviation (d), Maximum Loaded Resource (MLR), and Tasks assigned to MLR are identified.

Load balancing is an important factor to be focussed seriously as it keeps all the resources active; which means that all the resources should get utilized and hence task execution would be done in parallel fashion. When parallelism is found more then, it can be said that execution of tasks is done at a faster rate. Scheduling heuristics like Min-Min and Max-Min schedules different tasks to different resources efficiently and it is found that load balancing among the resources is not maintain properly. Therefore, it is focussed to incorporate load balancing in ETS algorithm.

Min-Min algorithm is found to select tasks with minimum execution time for allocation and due to this approach; tasks with longer execution times remain unassigned even when there is availability of resources. This causes the resources to remain in idle state for considerable period of times. The tasks are allocated to the grid resources with respect to minimum execution time. The load balancing method is applied on the previous allocation in order to assign certain task to other resources rather than continuing in the same resource as per the previous step. For the reallocation process, the ETS algorithm is designed to identify the Next Earliest Completion Time (NECT) of all tasks in MLR and the corresponding resources. Remove the first task from MLR and allocate it to the corresponding resource which produces the NECT. Then, calculate the makspan and mean square deviation (d); if the makespan is found less than the previous allocation then the tasks are reassigned on to the resource that are identified. If the value of d is closely related to one, then the optimal load balanced is achieved. Otherwise the previous step is repeated until it finds a minimum makespan and optimal load balance.

ALGORITHM 1

ETS ALGORITHM

---

1. Start
2. For all tasks T_i in meta-task Mv
3. For all resources R_j
4. $C_{ij}=E_j+r_j$
5. Do until all tasks in Mv are mapped
6. Assign task Tk to the resource Rl that gives the
    i. earliest completion time

7. Delete task Tk from Mv
   ii. Calculate the makespan , Aru, Ru, d.
8. Find MLR and their tasks
9. NECT(T_i) = Min (ET_{ij}) where R_j not already scheduled for T_i
10. Assign task T_k to the resource R_j
11. Verify the makespan with previous makespan & d ≈ 1
12. If $MK_{i+1} > MK_i$ & d value is tightly closed to 1 then stop the process
    iii. Else proceed to step 8
13. Stop

---

where,

Aru – Average Resource Utilization

Ru – Resource Utilization

MLR - Maximum Loaded Resource

NECT – Next Earliest Completion Time

d – Mean Square Deviation

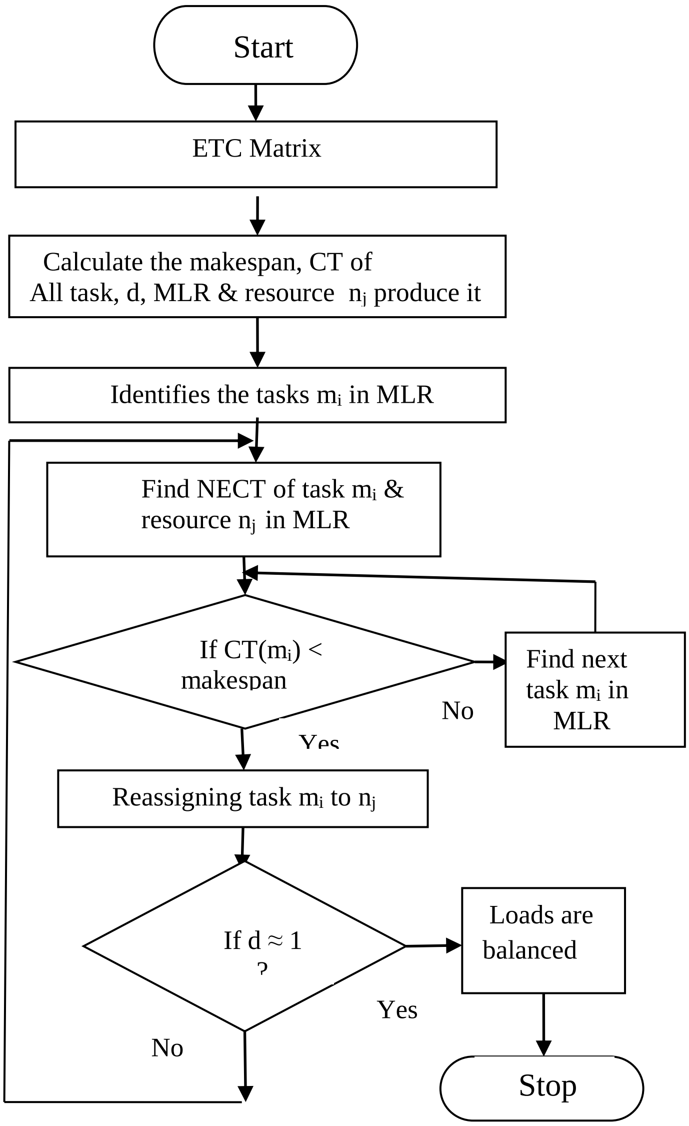

Figure 1. Methodology of ETS Algorithm

Here, the results are evaluated on the basis of following performance metrics.

Makespan: - Makespan is the measure of the throughput of the Grid. It can be calculated using equation (1). A better scheduling algorithm produces less makespan values.

$$\text{Makespan} = \text{Max}\,(CT\,(t_i, m_j))$$

$$CT_{ij} = RT_j + ET_{ij} \qquad (1)$$

Where,

CT completion time of resources,

ET_{ij} expected execution time of task 'i' on resource 'j',

RT_j ready time of resource' j'.

Average resource utilization rate: Average resource utilization rate of all resources can be computed as follows

$$Aru = \frac{\sum_{j=1}^{m} Ru_j}{m} \quad (2)$$

Here Ru_j is the resource utilization rate of resource r_j and m is the total number of resources.

Resource Utilization Percentage: Resource utilization Percentage of all resources can be computed as follows,

$$Ru_j = \sum_{i=1}^{n} \text{Where } t_i \text{ has been executed on } r_j \ (tf_i - ts_j) \quad (3)$$

Here, tf_i is the finish time and ts_i is the start time of task t_i on resource r_j.

Load Balancing Level: The mean square deviation of r_u is given by,

$$d = \sqrt{\frac{\sum_{i=1}^{m}(Aru - ru_j)^2}{m}} \quad (4)$$

5. EXPERIMENTAL RESULTS

Assume the grid environment with three resources. The execution times of the resources are shown in Table 1. Eight tasks *T*0 through T7 are in the meta-task *Mv* and the grid manager is supposed to schedule all the tasks within *Mv* on three resources *R0, R1* and *R2*.

Table 1: Sample ETC Matrix

ETC			
Resource / Tasks	R0	R1	R2
T0	3	4	7
T1	4	5	8
T2	9	3	10
T3	4	6	5
T4	7	19	18
T5	6	3	5
T6	6	7	4
T7	7	5	2

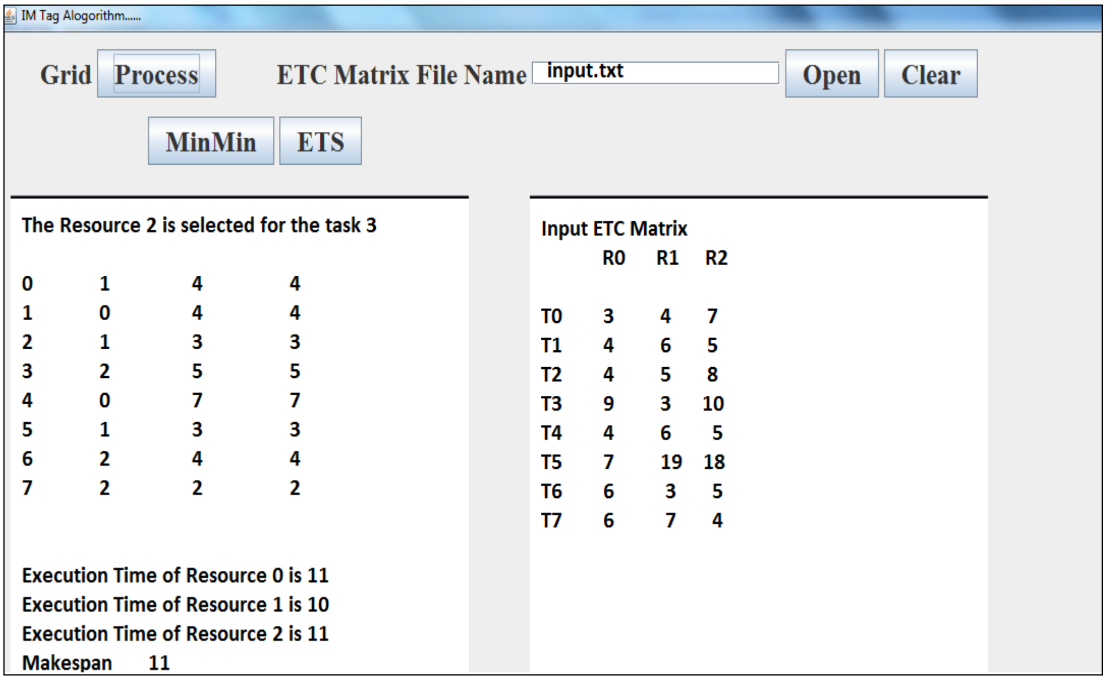

Figure 2. Execution of Min-Min Algorithm

Figure 2 shows the schedules of Min-Min algorithm. The completion time of R0 is 18 and R1, R2 results 6. From the above implementation, it is found that R0 is highly loaded and the remaining resources such as R1 & R2 are loaded less.

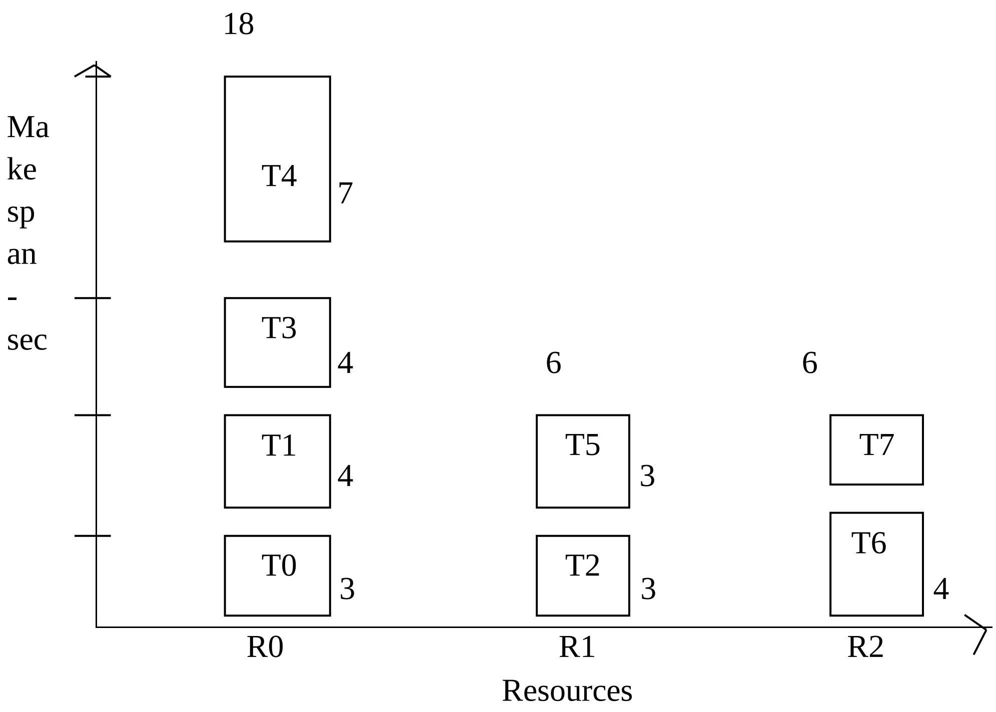

Figure 3: Chart of Schedules produced by Min-Min Algorithm

Figure 3 depicts the schedules of Min-Min algorithm on executing the tasks described in Table 1.

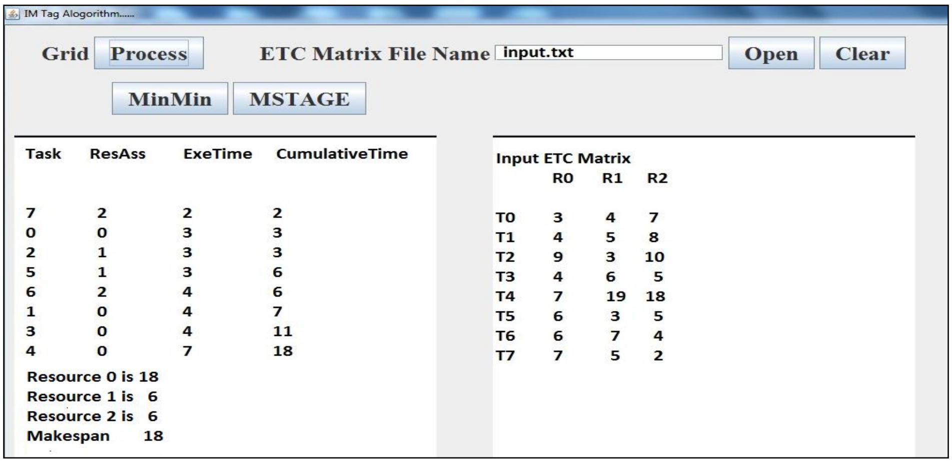

Figure 4. Execution of ETS algorithm

The execution time of proposed ETS algorithm is shown in Figure 4. The proposed algorithm ETS achieves a minimum makespan of 11 when compared to Min-Min Heuristic.

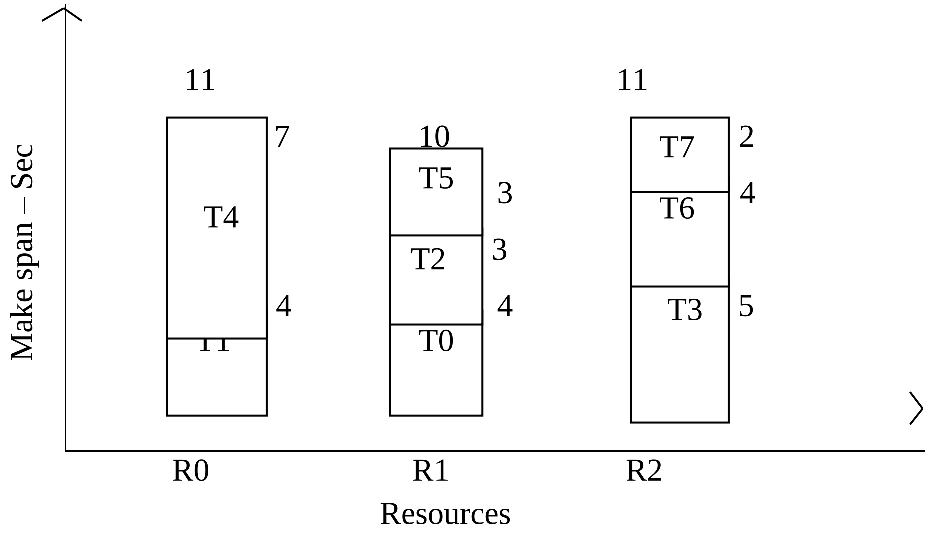

Figure 5. Gantt chart using ETS Algorithm for the Sample ETC Matrix

Figure 5 shows the schedules of proposed ETS algorithm which utilizes all the available resources. The following values are observed from the proposed, ETS algorithm:

Aru = 10.67

ru_j = { R0 = 11, R1 = 10, R2 = 11}

d = (0.9 ≈ 1)

The mean square deviation is 0.9, which is more or less equal to 1.0.

It is clearly observed from the results that the ETS algorithm utilizes all the resources effectively and produces less makespan value than Min-Min algorithm.

Improvement Percentage of Performance (IPP) can be calculated as:

$$IPP = (MSMM - MSETS / MSMM) * 100 \quad (7)$$

where MSMM is the Makespan of Min-Min and MSETS is the makespan of ETS.

Table 2

Problem Set	Min-Min	ETS	Improvement Percentage of ETS
4x2	9	7	22.22
8x3	18	11	38.89
20x10	42	35	16.67
30x10	68	60	11.77
50x10	96	84	12.5

Table 2 shows the various makespan for various ETC with various range of tasks and resources. It also shows the Improvement percentage of ETS algorithm compared to Min-Min heuristic.

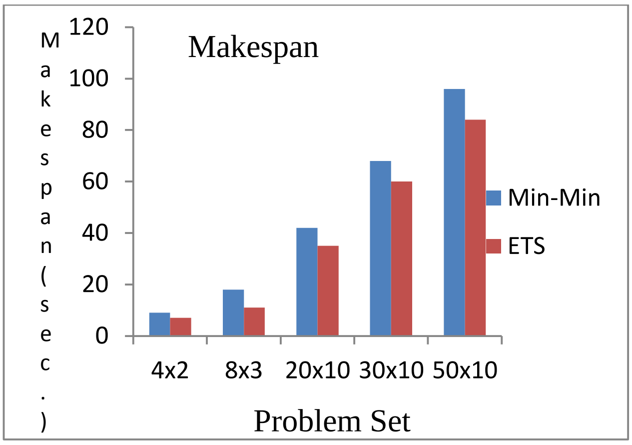

Figure 6. Makespan

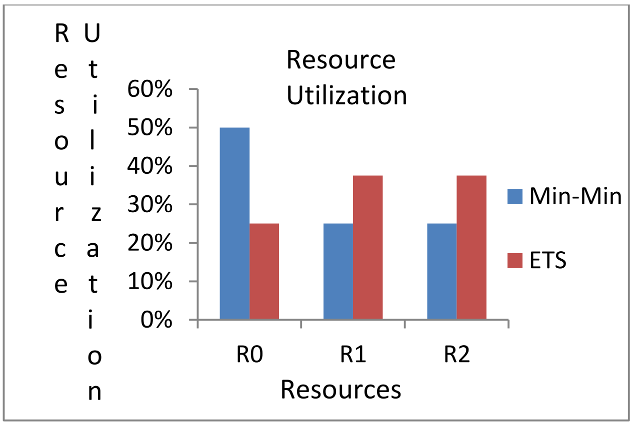

Figure 7. Resource Utilization

The graphical representation of the results between Min-Min and ETS in terms of makespan in Figure 6 and resource utilization ion in Figure 7 are the proof for the performance of the proposed ETS algorithm.

6. CONCLUSION AND FUTURE WORK

Traditional existing heuristic algorithms like Min-Min and Max-Min are applicable in small scale distributed systems. When the number of the little task is greater than the number of the bulky task, the Min-Min algorithm cannot schedule tasks, appropriately, and the makespan of the system gets relatively large and it does not provide a load balanced schedule. To overcome these limitations, a new task scheduling algorithm, ETS, is proposed. The experimental results obtained by the proposed algorithm for various problem sets shows that it outperforms the existing scheduling algorithms. This study is only concerned with the number of the resources and task execution time. The study can be further extended by taking into account of machine heterogeneity and task heterogeneity. Also, applying the proposed algorithm on actual grid environment and considering the cost factor can be other open problem in this region. In future, research will be directed towards the factors such as CPU workload, communication delay and so on.

REFERENCES

[1] Rajkumar Buyya and M. Murshed, “GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid Computing”, CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE Concurrency Computat.: Pract. Exper. 2002; 14:1175–1220 (DOI: 10.1002/cpe.710).

[2] Ian Foster, Carl Kesselman, Steven Tuecke, “The Anatomy of the Grid Enabling Scalable Virtual Organizations” International Journal of Supercomputer Applications, 2001.

[3] Siriluck Lorpunmanee, Mohd Noor Sap, Abdul Hanan Abdullah, and Chai Chompoo-inwai, “An Ant Colony

[4] Optimization for Dynamic Job Scheduling in Grid Environment”, World Academy of Science, Engineering and Technology 29, pp. 314- 321, 2007.

[5] Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. Basingstoke: Macmillan Higher Education.

[6] Florin Pop and Valentin Cristea. Intelligent strategies for dag scheduling optimization in grid environments. In *Proceedingsof the 16th International Conference on Control Systems and Computer Science (CSCS16’07), May 22-25, Bucharest,Romania, Printech, ISBN 978-973-718-743-7*, pages 98–103,2007.

[7] H. Singh and A. Youssef, “Mapping and scheduling heterogeneous task graphs using genetic algorithms”, in 5th IEEE Heterogeneous Computing Workshop (HCW '96), **(1996)**, pp. 86-97.

[8] He. X, X-He Sun, and Laszewski. G.V, "QoS Guided Minmin Heuristic for Grid Task Scheduling," Journal of

[9] Computer Science and Technology, Vol. 18, pp. 442-451, 2003.

[10] Sameer Singh Chauhan,R. Joshi. C, QoS Guided Heuristic Algorithms for Grid Task Scheduling, International

[11] Journal of Computer Applications (0975 – 8887), pp 24-31, Volume 2, No.9, June 2010.

[12] Dong. F, Luo. J, Gao. L and Ge. L, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping,"

[13] In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), IEEE, 2006.

[14] Etminani .K, and Naghibzadeh. M, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling," The Third IEEE/IFIP International Conference on Internet, Uzbekistan, 2007.

[15] Ranganathan, K. and Foster, I., "Decoupling Computation and Data Scheduling in Distributed Data Intensive

[16] Applications", Proceedings of the 11th IEEE Symposium on High Performance Distributed Computing (HPDC-11), Edinburgh, Scotland, July 2002.

[17] Ullah Munir. E, Li. J, and Shi. Sh, 2007. QoS Sufferage Heuristic for Independent Task Scheduling in Grid.

[18] Information Technology Journal, 6 (8): 1166-1170.

[19] Doreen Hephzibah Miriam. D and Easwarakumar. K.S, A Double Min Min Algorithm for Task Metascheduler on

[20] Hypercubic P2P Grid Systems, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 5, July 2010.

[21] Kamalam.G.K and Muralibhaskaran.V, , A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-

[22] Tasks On Heterogenous Computing Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.1, January 2010.

[23] Sadegh Nejatzadeh et. al., 2013, A New Heuristic Approach for Scheduling Independent Tasks on Grid Computing Systems, International Journal of Grid and Distributed Computing, 6: 97-106.

[24] GaurangPatel, Rutvik Mehta, Upendra Bhoi, 2015, Enhanced Load Balanced Min-Min Algorithm for Static Meta Task Scheduling in Cloud Computing, 3 rd International Conference on Recent Trends in Computing 2015 (ICRTC - 2015) published by Elsevier.