Volume 3, No. 2, March-April 2012

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Task Duplication Based Efficient Multi-Objective Grid Workflow Scheduling Algorithm

Dr.D.I.George Amalarethinam
Director-MCA & Associate Professor of
Computer Science, Jamal Mohamed College
Tiruchirappalli , Tamilnadu, India
di_george@jmc.edu

F.Kurus Malai Selvi*
Assistant Professor of Computer Science
Government College for Women,
Kumbakonam, India
fkmsjoe@gmail.com

Abstract: Grid computing facilitates global infrastructure for user to consume the services over the network. To achieve the promising potentials of enormous distributed resources, effective and efficient scheduling algorithms have to be used. Most of the parallel applications in grid computing fall into interdependent task model called workflow. Grid workflow scheduling represented by Directed Acyclic Graph (DAG) is a typical NP-Complete problem. The performance of the workflow scheduling is based on efficient scheduling. Scheduling is a process that maps and manages the execution of interdependent tasks on the distributed resources. In this paper, a new algorithm, named *Task Duplication Based Efficient Multi-Objective Scheduling (TDB-EMOS)* algorithm is proposed to satisfy multi objective functions. It maximizes the *resource utilization* in a grid, minimizes *makespan* and *communication cost*/*time* by reserving the resources in advance and schedules the task on priority. The proposed algorithm has been implemented with arbitrary task graphs and application graphs in a simulated environment. The results are compared with the well known *Min-Min, HEFT, EDOS (Efficient Dual Objective Scheduling)* and EDS-G (Economical Duplication Scheduling in grid) scheduling algorithms and showing that the proposed algorithm is yielding better results.

***Keywords*:** Grid Computing; DAG; Workflow scheduling; Heterogeneous systems; Parallel processing.

I. INTRODUCTION

The goal of grid computing is to make the illusion of a simple and powerful self managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. The term Grid comes from the electricity grid. Grids enable the sharing, selection and aggregation of a wide variety of resources including super computers, storage systems, data sources and specialized devices that are geographically distributed and owned by different organizations for solving large-scale computational and data intensive problems in science, engineering and commerce. Many of the large scale scientific applications executed on present-day grids are expressed as complex e-science workflows [1][2]. A workflow is a set of ordered tasks that are linked by data dependencies [3]. Workflow scheduling is one of the key issues in the workflow management [3].

The objective of scheduling is to minimize the completion time of a parallel application by properly allocating the tasks to the resources. Scheduling allocates suitable resources to workflow tasks so that the execution can be completed to satisfy objective functions imposed by users. In a grid environment, new challenges are the heterogeneity of resources and dynamic changes of several parameters of the environment like availability of resources and transfer time between nodes. Therefore advance reservation mechanisms are introduced to provide some guaranteed performance to users. Advance reservation has been identified as a key technology in order to guarantee that enough resources are available for time-critical workflow execution [4].

Workflow scheduling may be computation intensive or communication intensive (transaction-intensive). Transaction-intensive grid workflows are gaining more and more attentions with the prosperity of e-business and e-government applications. In the computation intensive workflow, as the communication cost is less, and it will not affect the performance of the workflow. But in the communication intensive workflows, the dependency between the tasks will affect the overall performance of the workflow.

Most of the scheduling algorithms are mainly focusing on optimization criteria namely *scheduling length* or *makespan* (i.e. the total completion time of a job). The proposed algorithm concentrating multi objective functions with the help of *Rank Function (RF)*, advance reservation of resources and duplication of tasks. Duplication based scheduling algorithms are designed to avoid communication time of results between tasks and to reduce the *makespan*.

The remainder of this paper is organized as follows; .Section 2 presents the problem statement and defines the term used in the proposed algorithm. The proposed *TDB-EMOS* algorithm is described in Section 3. Section 4 compares the results of proposed algorithm with existing scheduling algorithms. Section 5 concludes the work.

II. PROBLEM DEFINITION

A parallel program can be represented by a Directed Acyclic Graph (*DAG*) [9] $G = (V, E)$, where V is a set of v nodes and E is a set of e directed edges. A node in the DAG represents a task (A task is an indivisible unit of work) which in turn is a set of instructions which must be executed sequentially without preemption in the same resource. The weight of a task t_i is called the computation time and is denoted by $w(t_i)$. The edges in the DAG, each of which is

denoted by (t_i, t_j), correspond to the communication messages and precedence constraints among the tasks. It is a predecessor of t_i and t_j is a successor of t_i, that is, $t_i < t_j$ iff $e_{ij} \in E$. The weight of an edge is called the communication cost/time of the edge and is denoted by $c(t_i, t_j)$.

A sample *DAG* is given in Figure 1. The source task of an edge is called the parent task while the sink task is called the child task. A task with no parent is called an entry task and a task with no child is called an exit task. A child task can be carried out only to receive all messages of its parent tasks. When a task and its successor tasks are scheduled to the same resource, the communication cost is zero.

In the proposed algorithm three objective functions are considered, namely, maximizing the resource utilization, minimizing the total completion time (makespan) and communication cost of a job. Formally makespan can be defined as:

Minimization of makespan:

$$Min\{\max FT(t_i)\}$$

where $FT(t_i)$ is the finish time of task t_i.

Maximizing the resource utilization of the Grid system is another important objective. The execution time and idle time of a resource are known from the scheduled list is used to calculate the utilization of resources. *Resource Utilization (RU(R_i))*[8] of resource R_i is calculated as

$$RU(R_i) = \frac{\sum_{j=1}^{k} w(t_j, R_i)}{processor - speed}$$

where $\sum_{j=1}^{k} w(t_j, R_i)$ is the sum of all allotted task's computation time of resource R_i., *processor-speed* is the computation speed of the processor.

The *Average Resource Utilization (ARU)*[8] of all resources gives the overall utilization percentage of the grid resources is specified as:

$$ARU = \frac{\left(\sum_{i=1}^{p} RU(R_i)\right)}{p} * 100$$

where p is the total number of resources.

The only way to minimize the communication cost is to schedule a task and its successor tasks to the same resource. The main idea behind duplication based scheduling is to utilize processor idling time to duplicate predecessor tasks. This may avoid transfer of results from a predecessor task, through a communication channel to the task. This may eliminate waiting slots on other processors. Some of these schemes are found in [5][6][7].The duplication is carried out until system resources are exhausted.

A. Terminologies:

The proposed algorithm is introducing task duplication on *EDOS* algorithm [8]. It shows that duplication on EDOS yielding better result than EDOS algorithm. This algorithm prioritizes the tasks to be scheduled on the basis of a value computed by a *Rank Function (RF)* as same as in *EDOS.* The *RF* is calculated for each task by level-wise is explicitly specified the i^{th} level in j^{th} task. To compute *RF* the *Ratio of Actual Communication Time (RACT)* and computation time of each task is required.

$$RF(t_{ij}) = RACT(t_{ij}) * w(t_{ij})$$

Where *i* refers to number of levels and *j* refers to the number of tasks in a level

$$RACT(t_{ij}) = \frac{\max(c(pred(t_{ij}), t_{ij}))}{LCT(i)},$$

Where t_{ij} is the i^{th} level in j^{th} task, $max(c(pred(t_{ij}), t_{ij}))$ is maximum communication time between

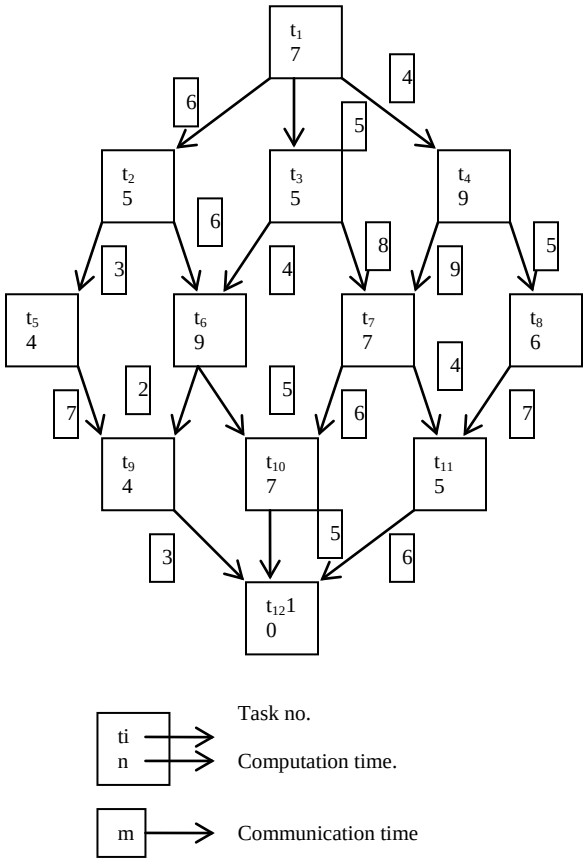


Figure1. A sample DAG

the $pred(t_{ij})$ tasks *and* task t_{ij}, that is, the edge between all parent tasks of t_{ij}. $RACT(t_{ij})$ is calculated for *n* number of tasks, *Level-wise Computation Time LCT(i)* is calculated as follows

$$LCT(i) = \sum_{k=1}^{l(i)} c(pred(t_{ik}), t_{ik}), \forall i = 1, 2, .. m$$

number of levels,

where *l(i)* is the total number of task in a i^{th} level.

The *Approximate Computation Time (ACT)* of task t_{ij} is the actual time required to execute the task in resource r_k is expressed as

$ACT(t_{ij}, r_k) = w(t_{ij})/speed\ of\ resource(k).$

In the parallel applications the performance of parallelism is evaluated based on *speedup* parameter. *Speedup* is commonly defined as the ratio between the total elapsed time on one processor divided by the total elapsed parallel

execution time on all processors. Based on this definition the speedup ratio is calculated.

Speedup ratio computation requires the *Ideal Parallelism (IP)* and *Equivalent Number of Resources(ENR)*. In heterogeneous systems the total number of resources is not equal to as such as in homogeneous systems. In homogeneous system all resources are same type and speed, but in heterogeneous all are different. To evaluate the parallel performance, the actual number of resources is calculated based on their speed. i.e.*ENR.*

Ideal Parallelism(IP) is computed as the Job Completion Time On Single Resource(*JCTOSR)* is divided by *Equivalent Number of Resources(ENR).*

$$IP = \frac{JCTOSR}{ENR}$$

The *IP* is Ideal parallelism i.e. hundred percentage parallelisms. In real situation hundred percentages is not possible. But how much closer to *IP* is evaluated. Speedup ratio is calculated using *IP* and *makespan*.

$$SpeedupRatio = \frac{IP}{Makespan}$$

This *speedup ratio* formula is used to evaluate the parallel performance of both homogeneous and heterogeneous systems.

III. A *TDB-EMOS* ALGORITHM

An *EDOS* algorithm produces minimum makespan and maximum resource usage but it doesn't consider the communication cost as its optimization criteria. The resources are reserved in advance therefore all resources are available from the job starting time to finishing time. The resources should be utilized properly and utilizing idle time slot for duplication to reduce the starting time of successor tasks. After duplication the communication time is reduced and it also fulfilled the other optimization criteria too.

The proposed *TDB-EMOS* has three objective functions. To achieve all these objectives three factors are to be considered. Makespan is always bothering the finishing time of each and every task of a workflow. The resource utilization is selecting the best resource for a task and communication time is depending on the precedence constraints.

This algorithm is similar to an *EDOS* except the duplication. Initially the entry task is duplicated as much of its number of successors, if the starting time of its successor task is reduced. Intermediate task duplication is based on which task has the maximum number of task to travel to reach the exit task.

A. The TDB-EMOS Algorithm:

Procedure TDB-EMOS Algorithm

a. {
b. Calculate RF
c. Reserve m number of resources from the grid resource broker & Place it in Resource Available List (RAL)
d. While all tasks are not scheduled do
e. {
f. RTQ ←t_i when parent tasks finished
g. Sort RTQ in an ascending order of RF
h. While (RTQ != empty) do
i. {
j. If pred(t_i)=0 then // If t_i is an entry task
k. If duplication of t_i can reduce ACT(Succ(t_i)) then
l. Duplicate t_i in No. of Succ(t_i) resources
(No. of times duplication = No. of Succ(t_i))
// The duplication of task is in idle time slot of resources Using insertion based approach//
m. Compute the Minimum Finish Time of t_i in R_j resources
n. R_j ← t_i //Schedule t_i on resource R_j
o. If (No. of resources in RAL is Even) then
p. Max RF(t_i) to be selected to schedule
q. Else
r. MinRF(t_i) to be selected to schedule
s. endif
t. If duplication of t_i can reduce ACT(Succ(t_i)) (or) t_i has more tasks to reach exit tasks then
u. duplicate t_i with idle time slots of available resources
a. R_j ← t_i
b. }
c. }
v. }

IV. RESULTS AND DISCUSSION

The simulation is performed on an Intel Pentium IV personal computer. A Java NetBeans IDE based simulation program to generate DAG has been developed to simulate this experiment. Sample run of the given task set is shown in Figure 2.

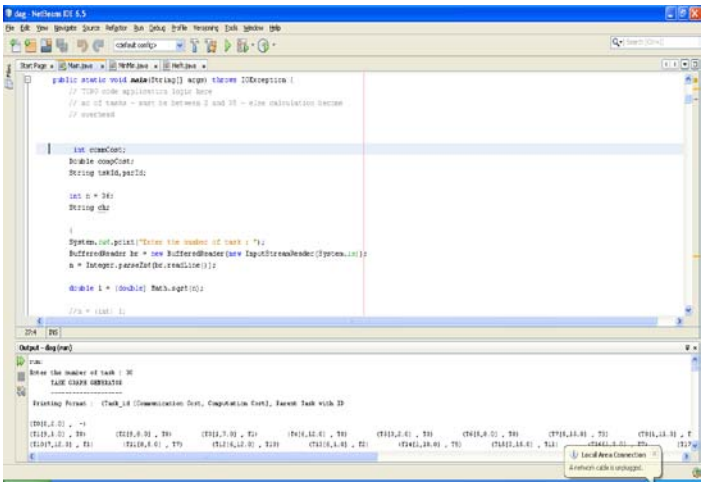


Figure 2. Sample Run of a given task set

For the simulation study the number of tasks in the arbitrary task graphs considered is *20, 25* and *30*. The maximum number of resources is reserved in advance at most the maximum number of tasks in any level of a DAG. Therefore the number of resources reserved is *4, 6* and *8* to favor all specified tasks. A resource is a basic computational device or service where tasks, jobs and applications are scheduled, allocated and processed accordingly. Resources have their own characteristics such as CPU characteristics, memory, etc. One of the CPU characteristic is the speed of the

resource considered for this simulation. The speed is varied as *1, 1.25, 1.5* and *1.75*.

The experiment is conducted for each set of tasks, that is, fixed number of tasks with *4, 6* and *8* resources having different speed with *Min-Min*, *HEFT*, *EDOS* and Economical Duplication Scheduling algorithm in Grid (EDS-G)[9] algorithm separately. The possible combinations of experiments are conducted with fixed number of tasks while varying number of resources and vice-versa. The results are tabulated after conducting the experiments.

In grid scheduling the three metrics, namely, *makespan*, *communication cost* and *resource utilization* and one more metric *speedup ratio* i.e. performance of parallelism parameter are considered for comparison.

A. Makespan:

The main performance measure of a scheduling algorithm is the finishing time of a last task is *makespan*. The Figure 3.a, Figure 3.b and Figure 3c shows that almost all cases the makespan of proposed TDB-EMOS algorithm is minimized.

B. Communication cost:

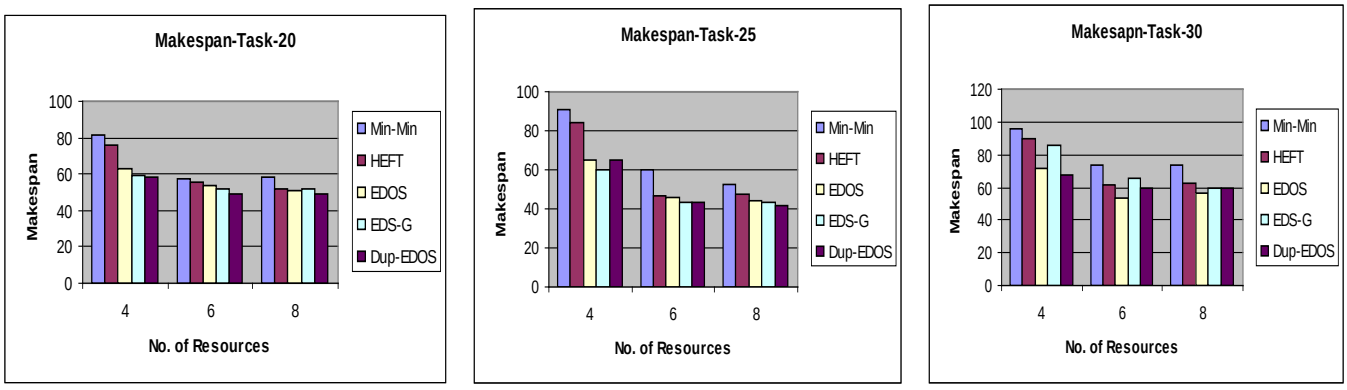
The communication cost is depends on the number of tasks with different number of resources. The communication cost is minimized in 20 tasks with 8 resources proves in Figure 4.a., 25 tasks with 6 resources as shown in Figure 4.b, 30 tasks with 6 resources and 30 tasks with 8 resources displays in Figure 4.c

C. Resource utilization:

The resources, which have been reserved in advance, are available till the execution of the last task. As shown in Figure 5.a, Figure 5.b and Figure 5.c the *TDB-EMOS* algorithm has utilized resources better than *Min-Min*, *HEFT*, *EDOS* and *EDS-G* algorithms in all cases. This proves that all resources are effectively utilized.

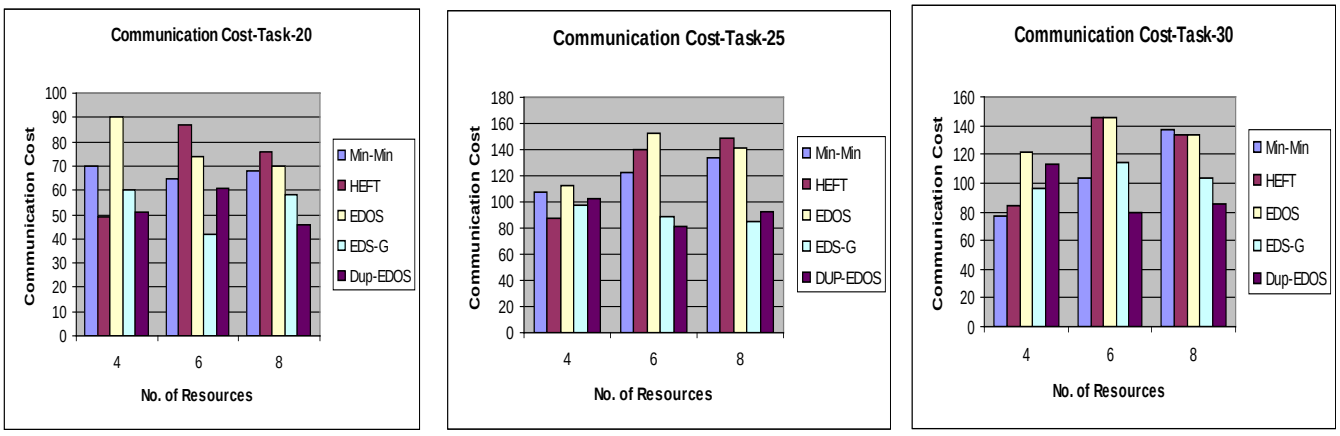
D. Speedup Ratio:

The *speedup ratio* is maximized in almost all cases of *TDB-EMOS* algorithm than *Min-Min*, *HEFT*, *EDOS* and *EDS-G* algorithms as shown in Figure 6.a, Figure 6.b and Figure 6.c. The *speedup ratio* indicates how well a system is using its potential power.



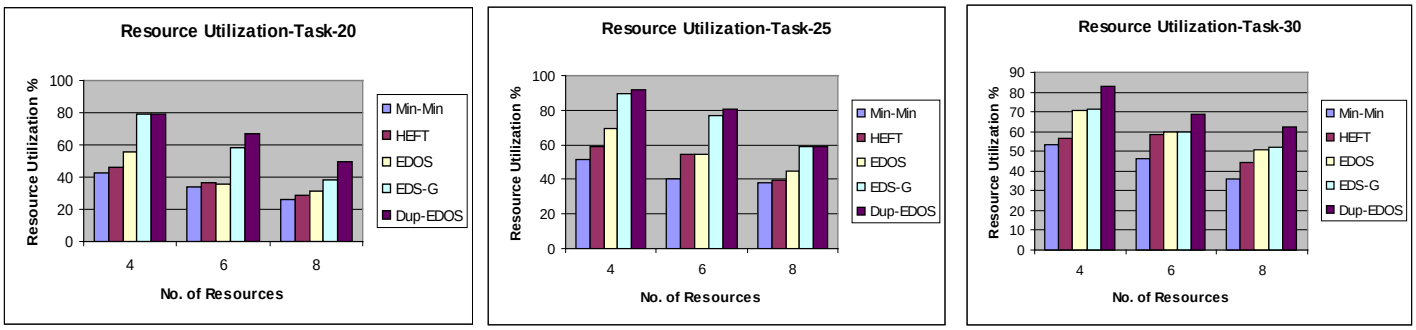
a. Makespan - 20 tasks b. Makespan - 25 tasks c. Makespan - 30 tasks

Figure 3. Comparative study of Makespan with variousnumber of resources with different algorithms



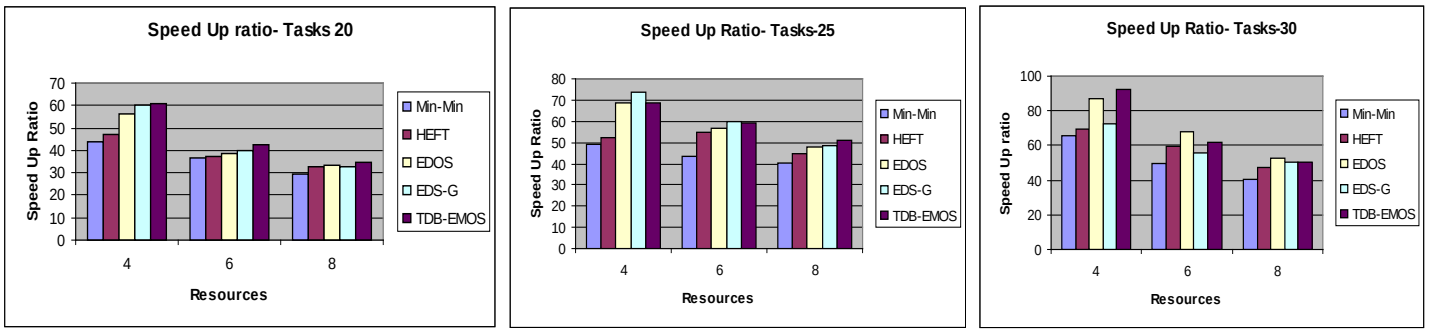
a. Communication n.cost.-20 tasks b. Communication cost- 25 tasks c. Communication. cost - 30 tasks

Figure 4. Comparative study of Communication cost with various number of resources with different algorithms



a. Resource. Utilization. –20 tasks b. Resource. Utilization. –25 tasks c. Resource. Util.ization – 30 tasks

Figure 5. Comparative study of Resource Utilization with various number of resources with different algorithms



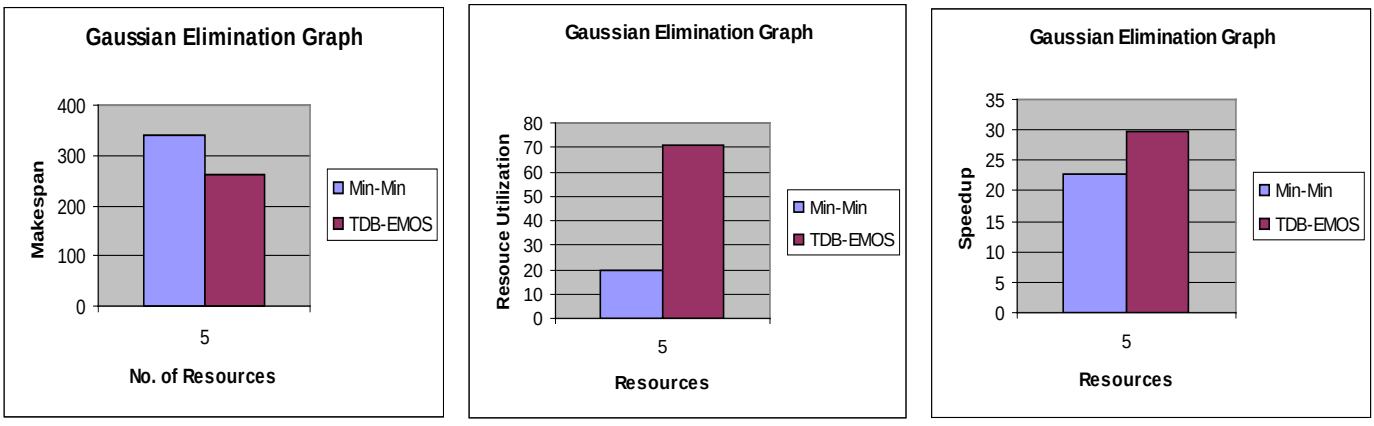
a. Speed up - 20 tasks b. Speed up - 25 tasks c. Speed up -30 tasks

Figure 6. Comparative study of Speed up Ratio with various number of resources with different algorithms

E. Application Graph:

Most of the Gaussian Elimination graph is a representation of communication intensive problem. The list scheduling algorithms are not suitable to execute communication intensive problem in a parallel mode. It is suitable to execute in a sequential manner. The communication intensive problems are well suited to adopt duplication based algorithm.

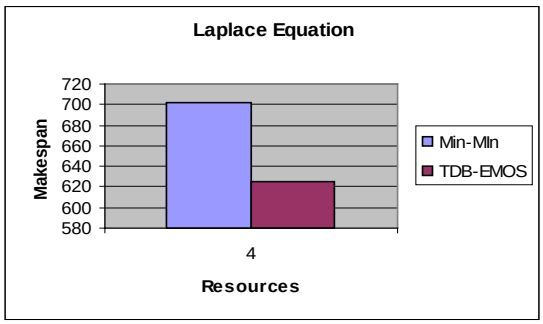
The Figure 7a illustrates that the *TDB-EMOS* algorithm of Gaussian Elimination graph minimizes the *makespan* and maximizes the resource utilization as shown in Figure 7b and speed up ratio as displayed in Figure 7c. The *Min-Min* algorithm utilized only one resource and other 4 resources were idle but the proposed algorithm utilized all 5 resources effectively. In both algorithms the communication cost is zero.



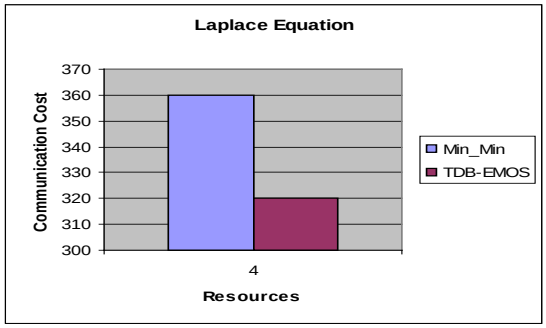
a. Makespan b. Resource utilization, c. Speed up ratio

Figure 7. Comparative study of Makespan, Communication. Cost and Resource Utilization with Min-Min algorithm

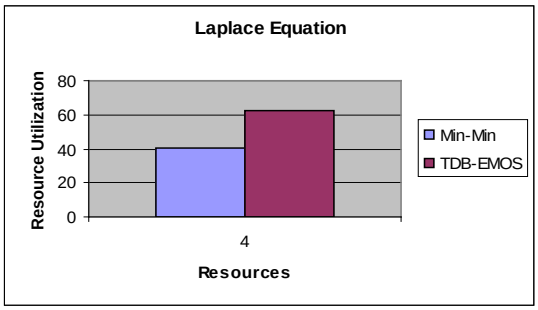
In Laplace Equation both *Min-Min* and *TDB-EMOS* algorithms are using 4 resources. Figure 8a and Figure 8b show that the *makespan and* communication cost of *TDB-EMOS* are less than *Min-Min* algorithm. The resource utilization and speed up ratio of *TDB-EMOS* are higher than *Min-Min* algorithm as displays in Figure 8c and Figure 8d.



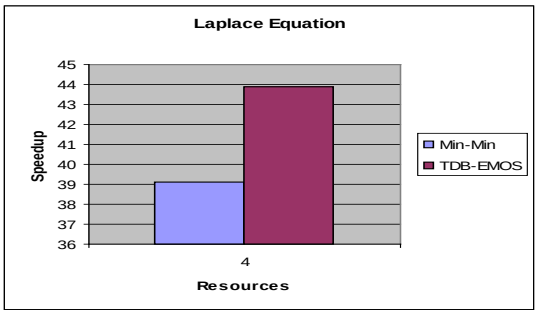
a. Makespan



b. Communication Cost



c. Resource utilization



d. Speed up ratio

Figure 8. Comparative study of Makespan, Communication. Cost, Resource Utilization and Speed up ratio with Min-Min algorithm

V. CONCLUSION

Grid scheduling is one of the potential area of research, which has to be resolved by grid researchers. It is a complex problem, which aims to map existing tasks to accessible storage and computational resources in order to get their effective utilization. Scheduling is the decision process by which application components are assigned to available resources to optimize various performance metrics. The *TDB-EMOS* algorithm minimizes the *makespan* and *communication cost* and maximizes the *resource utilization* and *speedup ratio*. The same *speedup ratio* formula is used for both homogeneous and heterogeneous systems. The exhibited figures have demonstrated that the *TDB-EMOS* algorithm satisfies the multi objective functions. The experiment has been conducted with arbitrary DAGs. To validate the performance of the proposed approach communication intensive task graphs like Gaussian Elimination method and Laplace transformation method are used. This algorithm can be used for task graphs with large number of tasks; still better results can be expected.

VI. REFERENCES

[1]. C. Laity, N. Anagnostou, B. Berriman, J. Good, J. C. Jacob, and D. S. Katz, "Montage: An Astronomical Image Mosaic Service for the NVO", Astronomical Data Analysis Software & Systems (ADASS) XIV, Pasadena, California, October 2004.

[2]. P. Blaha, K. Schwarz, G.K.H. Madsen, D. Kvasnicka and J. Luitz, "WIEN2k, An Augmented Plane Wave + Local Orbitals Program for Calculating Crystal Properties", Karlheinz Schwarz, Techn. Universität Wien, Austria, ISBN 3-9501031-1-2, 2001.

[3]. J.Yu, R.Buyya , " A Taxonomy of Workflow Management Systems for Grid Computing", Journal of Grid Computing 3(3-4), pp. 171–200, 2005.

[4]. L.-O. Burchard, M. Hovestadt, O. Kao, A. Keller, and B. Linnert, "The Virtual Resource Manager: An Architecture for SLA-aware Resource Management," in 4th Intl. IEEE/ACM Intl. Symposium on Cluster Computing and the Grid (CCGrid), Chicago, USA, 2004.

[5]. C. Papadimitriou and M. Yannakakis, "Towards an architecture independent analysis of parallel algorithms," SIAM Journal on Computing, Vol. 19, pp. 322-328, 1990.

[6]. J. Y. Colin and P. Chretienne, "C.P.M. scheduling with small computation delays and task duplication," Operations Research, pp. 680-684, 1991.

[7]. G. D. Li, D. X. Chen, D. M. Wang, and D. F. Zhang, "Task clustering and scheduling to multiprocessors with duplication," in Proceedings of the International Parallel and Distributed Processing Symposium, pp. 6b, 2003.

[8]. D.I.George Amalarethinam, F.Kurus Malai Selvi," An Efficient Dual Objective Grid Workflow Scheduling Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 33– No.1,pp.7-12, November 2011.

[9]. Amit Agarwal, Padam Kumar, Economical Task Scheduling Algorithm for Grid Computing Systems ,Global journal of Computer Science and Technology, Vol. 10 Issue 11 (Ver. 1.0).pp.48-53 October 2010.