ISSN:0975-9646
Leyli Mohammad Khanli et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1300-1304

A New Algorithm for Scheduling Parallel Tasks in Utility Grids using Reserve Resources for Tasks with Deadline

Leyli Mohammad khanli#, Motahareh Ahmadi*

#Computer Department, Tabriz University
Tabriz university, Tabriz, Iran
*Computer Department, Azad University of Tabriz
Azad University of Tabriz , Tabriz, Iran

Abstract— **In recent years, Grid computing systems have emerged as a solution to achieve distributed systems. Grid System is a collection of computing resources and users that are scattered around the world. These systems are developing and becoming more widespread with ever-increasing speed. Development of Grids and increase of the number of available resources and also increase in the number of users' requests to perform their computing tasks with minimum cost and in the least possible time, have made the issue of resource allocation and their scheduling as a challenge in such systems. On the other hand, some of users' requests may have deadlines and this issue makes scheduling problem more critical. In this paper for the first time, we have proposed a method for tasks scheduling using some reserve resources, which in addition to considering minimization of time and costs, it also considers tasks deadline. The performance evaluation is conducted using MATLAB software and is compared by MinCTT method. We have shown that in addition to performance improvement for tasks which have deadline, less time complexity shall also be obtained.**

Keywords— **Parallel applications, Resource Reservation, Resource Scheduling, Utility Grids, Resource Management.**

I. INTRODUCTION

Grid System is one of the important topics discussed currently in computer world. These systems have altered as a solution to respond the increasing computing demands of users who are scattered around the world. The Grids respond their demands using idle resources application of users who are scattered all over the world. Another important property of these resources is their dynamism.

Since Grids are not limited in terms of geographic coverage, therefore their available resources are geographically distributed and are probably heterogeneous [1]. Also, because the resources are located in different sites and each site belongs to various organizations and institutions, different management policies are governing on sharing of and access to resources [1]. On the other hand, Grid environment is a dynamic and optional one, meaning that in every moment a number of resources are inaccessible and some others are available [1]. Bearing in mind the mentioned issues, the effective resource management is considered a vital issue in Grid systems [1]. Moreover, scheduling in such systems must be done efficiently so that on the one hand the users and applications can receive services to its best and on the other hand optimal and maximum utilization can be obtained from all Grid resources [1]. The problem emerges when users in such environments prefer to pay the lowest cost towards the services which they receive. Besides, they tend their tasks to be done in the least possible time [3].This issue is more acute when users' requests have deadlines, that is to say the performance of their tasks should necessarily be terminated within given time interval, otherwise the performance of the task has no significance for users anymore (hard deadline) or is less significant (soft deadline).

In this paper, we have proposed an approach for tasks scheduling, that in addition to considering time and cost priorities on behalf of users and minimizing them, takes into consideration tasks deadlines as well [2]. In this method, we use a series of reserve resources for scheduling the tasks that have deadlines and we schedule the rest of tasks on remainder resources.

Furthermore, we initially present some related instances carried out in this field. Then we explain the mechanism of Meta-scheduler and its time complexity using our proposed algorithm. Performance evaluation of our method and conclusion constitute next sections of the paper.

II. RELATED WORKS

Commercial models in Utility Grids are categorized in two sectors; market models and auction [2]. GCommerce is an economic research to valuate Grid resources which indicated that the auction model does not function well in Grid [6]. Accordingly in this work we have focused on market commercial models in Grid and compare our work with other market commercial models [2]. Feng elsewhere [7], has proposed an algorithm for scheduling of task related requests that optimizes cost and time by considering budget and deadline constraints. In this work, no solution has been predicted for conformation the contradiction of simultaneous users of resources [2]. Munir elsewhere [8], has proposed QOS Sufferage algorithm for scheduling independent tasks in Grids. This work focuses merely on improving the total time and considers no trade-off between cost and time [2] and does not satisfy the deadline constraints of tasks. Kumar elsewhere [9], has provided two algorithms named HRED and HRED-T in which he has only considered cost reduction and has not considered a way for time improvement and deadline constraints. Buyya elsewhere [10], Gridbus Broker and Abramason elsewhere [11], have proposed Nimrod/G. These systems act toothily for scheduling Sweep Parameter requests with deadlines and cost constraints, and a solution has not been considered for users' different QOS requirements and their deadline constraints. Kumar and Buyya elsewhere [2], have suggested three scheduling methods named MinCTT, MaxCTT and SuffCTT which perform trade-off management between cost and time for scheduling parallel tasks in service Grids. Although this work tries to optimize the time of tasks performance, but does not necessarily perform them in their deadlines.

Some measures have been already carried out in the area of scheduling with the possibility of Advance reservation (e.g. [12]), but these methods are different from our work. Advance reservation are already implemented by the user to carry out the tasks user prefers, but in our work resources reservation is performed by Meta-scheduler and to execute the tasks that have deadline.

III. Performance of Meta-Scheduler

A. Meta-broker

The Meta-broker which we have used in this paper is considered as the future commercial model [4, 5].

As you observe in Figure 1, three parts are involved in this economic system: service provider, users and Meta-broker [2]. Detailed explanations are given in this regard in Kumar's paper [2] but a brief explanation regarding service providers which include resources; these services should perform requests that receive from the local users in any location and requests that come through Meta-broker. In this respect, they prepare some information regarding availability and costs of using processors per second in given intervals and present them to Meta-broker [2].

The users submit their requests to Meta-scheduler in order to perform on resources [2]. The users are willing to perform their requests in the shortest time and with the minimum cost [2]. Therefore, they submit a trade-off factor which represents the priority of cost over time for them [2]. This factor can be set by any of the users or by Meta-broker for all requests [2]. The Meta-broker receives collected data by providers and users and based on these information allocates resources time intervals to users' requests [2]. Scheduling of users' requests is done in Batch mode and at the end of each time interval [2]. The aim of Meta-broker is to schedule all users' requests on idle resources time intervals in the shortest possible time and with the minimum cost [2].

In this paper, as you will see in the following sections, we have used this model and at the same time we have tried to improve Kumar's method [2] to in addition to minimize time and cost in resource allocation, it can consider deadline constraints of tasks as well.

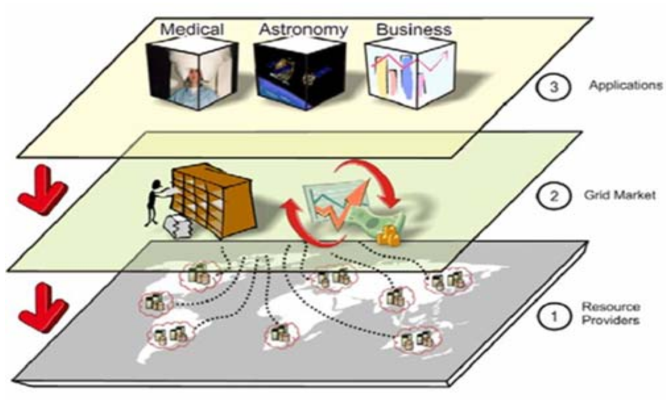


Figure (1): Meta-broker System [2]

The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

Recommended font sizes are shown in Table 1.

B. Problem statement

Since users have two major quality of service needs which include minimizing total execution time and minimizing costs of using resources [3], elsewhere [2] a cost standard formula (Relation 1) is presented in which the user by providing a trade-off factor specifies cost priority over time in resources allocation. Namely:

$$\phi(i,j,t) = \delta \frac{c(i,j)}{\gamma_i} + (1-\delta)\frac{\alpha(i,j)}{\beta_i} \qquad (1)$$

Δ In this formula δ is trade-off factor which is determined by the user. c (i, j) is expendable cost to perform request i on resource j. γ_i is the average cost of performing request i. α (i, j) is as response time of request i on resource j and βi is the average running time of request i on resource j which is determined based on ETC (Estimate To Completion) time to execute request i on resource j.

After calculating the standard cost of performing a request on all intervals of entire resources, the minimum of them will be selected to perform the request [2].

Subsequently Kumar [2] has proposed 3 solutions using this formula, including:

MinCTT, MaxCTT and SuffCTT. In this paper, we use MinCTT method as an example.

In our proposed solution, Meta-broker for choosing reserve section resources considers factors such as stronger processors and resources which are not ill repute (the resources that within a specified time interval could not be able to complete more than n number of requests before the preferred deadline). Obviously, the cost of these resources is higher than other resources.

C. Our Proposed Algorithm

In this work, we categorize resources to two sectors of reserve and non-reserve resources (Figure 2). Reserve resources are considered for tasks which have deadline and non-reserve resources are considered for other tasks.

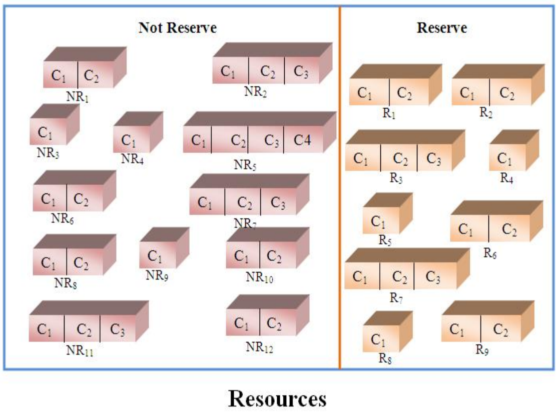


Figure (2): Resource Distribution

Since scheduling takes place in batch, the users send their requests to Meta-broker during each time interval and at the end of each time interval the Meta-broker reviews and schedule submitted requests (Figure 3).

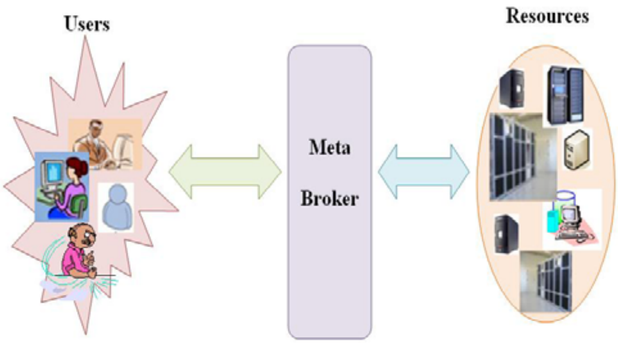


Figure (3): The way Meta-broker interacts with resources and users

The Meta-broker acts in this way:

The users submit their deadline and non-deadline requests to Meta-broker. The Meta-broker after receiving the requests, reviews them whether the requests have deadlines constraints or not? If there is no deadline constraint the request will be scheduled in non-reserve resources sector with one of the three methods MinCTT, MaxCTT, SufCTT [2] according to Meta-broker policy. If the request has deadline constraint, the Meta-broker reviews whether it can satisfy this constraint by allocating sufficient resources to submitted request from reserve resources or not? If it can perform it, it will send yes response along with the fee that user should pay and asks the user whether he/she is willing to run the task? Otherwise, it will send No response to user and asks whether he/she is willing to run the task in non-reserve resources or not. If user's response is positive, his/her task will be scheduled on appropriate resources.

Flowchart 1 shows scheduling of tasks using this method.

Advantages:

This work compared with methods such as MinCTT, MaxCTT and SufCTT [2], which do not regard deadlines, considers deadlines as well and this is a very important advantage for tasks with hard deadline. On the other hand, in this method it is for the first time that a set of reserve resources is considered merely for tasks with deadline the possibility of requests failing reaches minimum amount. Also it can benefit from advantages such as cost and time trade-off which is considered by user. On the other hand, since it performs this task only for requests that tend to be run within deadline, it will not impose high costs of expensive reserve resources to all tasks.

D. Time Complexity

Kumar elsewhere [2] has obtained a time complexity for MinCTT [2] method in the form of relation 2:=

$$O(n^2 \sum_{j \in R(t)} TS(j,i,t)) \qquad (2)$$

But in our method given that tasks and resources are divided into two sectors and the number of tasks which are scheduled by MinCTT [2] algorithm and the number of resources which are scheduled using MinCTT [2] method are less, therefore time complexity of algorithm will also be reduced.

In this way: If m is tasks with deadline and l is non-deadline tasks, we have relation 3:

$$n=l+m \qquad (3)$$

Also, if rr is reserve resources and r is non-reserve resources, we have relation 4:

$$R(t)=rr(t)+r(t) \qquad (4)$$

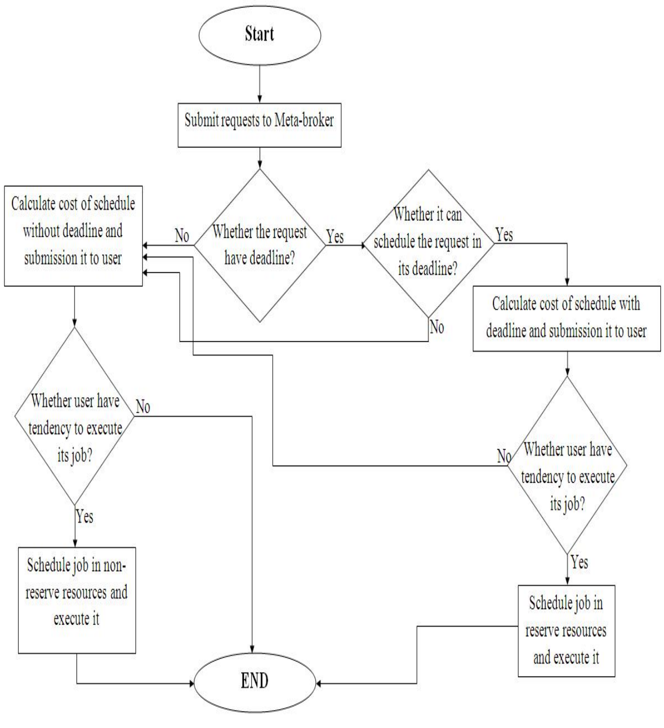
So according to relations 3 and 4, we have relation 5:

$$l<n\ ,\ r(t)<R(t) \qquad (5)$$

And according to relations 2 and 5 the complexity will be in the form of relation 6:

$$O(l^2 \sum_{j \in r(t)} TS(j,i,t)) \qquad (6)$$

Therefore, considering the above relations we proved that our method is less complex.



Flowchart (1): Scheduler Performance

IV. PERFORMANCE EVALUATION

Performance evaluation has been conducted using MATLAB software. As you can see in Table 1, we scheduled 10 tasks on 6 resources each of which has different submission and performance times. In this model three of the tasks have deadline (tasks 5, 8 and 10). Here we considered δ value equal to 0.25 in order to optimize the time further.

Table (1): The Scheduled Model

job	submission time	execution time	number of cpus required	Dead line
1	5	16	28	-
2	3	7	34	-
3	10	14	25	-
4	1	3	56	-
5	2	2	32	12
6	5	15	15	-
7	4	10	42	-
8	6	8	30	15
9	8	19	29	-
10	7	4	34	10

As Kumar's assessments [2] indicate MinCTT method in most cases functions better than MaxCTT and SuffCTT, we once scheduled the tasks by MinCTT [2] method. As you can see in Figure (4) the sequence of task performance on resources is as follows.

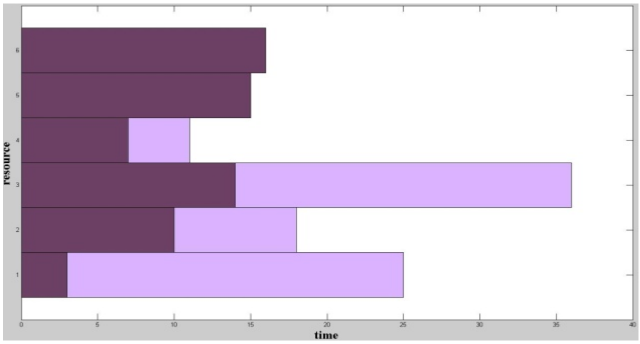


Figure (4): Tasks Scheduling on Resources using MinCTT [2] Method

As it is indicated in Figure 5 the tasks which have deadline are not conducted within their deadlines.

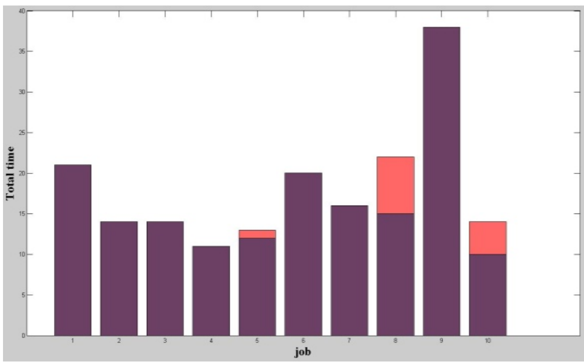


Figure (5): Tasks Performance Time using MinCTT [2] Method

Red color indicates the time which passed from tasks deadline.

Subsequently we scheduled the tasks with our proposed method. Namely we considered two of resources as reserve resources and scheduled the tasks which had deadline on these resources and scheduled other tasks with other resources using MinCTT [2]. This scheduling is presented in Figure (6) diagram.

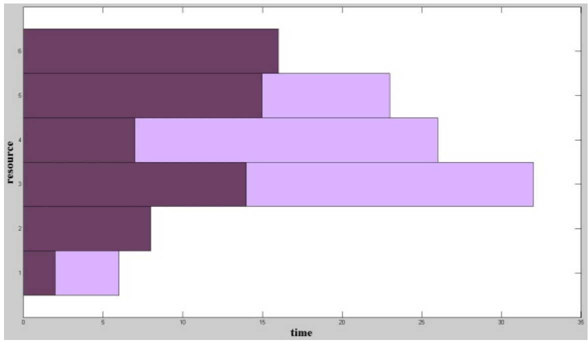


Figure (6): Tasks Scheduling on Resources by the Proposed Method

After reviewing tasks performance time, as it can be seen in Figure 7 the tasks with deadline were completed within their deadlines.

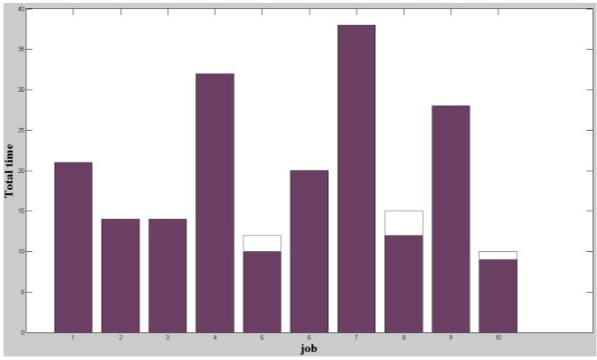


Figure (7): Tasks Performance Time using the Proposed Method

In Figure 7 white color represents the remaining time of tasks deadline.

In a comparison conducted between tasks performance time using MinCTT method and our proposed method (as you can see in the chart of Figure 8), the total execution time has not changed.

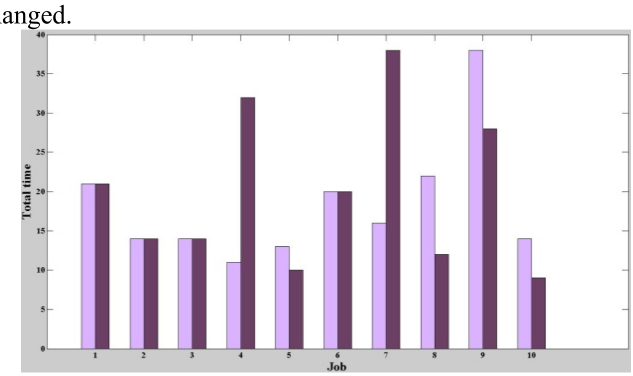


Figure (8): Comparison between Tasks Performance Time in MinCTT Method and our proposed Method

V. CONCLUSION

Today Grid has become a solution to increase the computing power of users. On the other hand, different service quality requirements of users have altered the resources scheduling and management to a challenge in these areas.

In this paper, we proposed a new scheduling method that in addition to considering different quality of service requirements of users, also takes into consideration deadline constraints of their tasks.

To evaluate the performance we compared this method with MinCTT [2] method. The assessments indicate that our proposed method in addition to advantages of this method has another advantage that considers tasks deadline constraints. The assessment shows that the total execution time will not increase via this method. Besides, we proved that time complexity of our method is less than MinCTT [2] method.

VI. FUTURE WORKS

In future, we can work on determining more precise criteria to specify reserve resources. Also algorithm can be implemented on a broader model of tasks and resources.

REFERENCES

[1] M.Fathi, F.Mehryari, *Grid computing and Different Applications,* Tehran, Noorpardazan Publisher, 2009.

[2] S.K. Garg ,R. Buyyaa , H. Jay Siegel, Time and cost trade-off management for scheduling parallel applications on Utility Grids, Future Generation Computer Systems, 2009

[3] J. Yu, R. Buyya, C. Tham, Cost-based scheduling of scientific workflow application on utility grids, in: Proceedings of the First International Conference on e-Science and Grid Computing, Washington, DC, USA, 2005.

[4] D. Neumann, J. Stoesser, A. Anandasivam, N. Borissov, SORMA-building an open grid market for grid resource allocation, in: Proceedings of the 4th International Workshop on Grid Economics and Business Models, Rennes, France, 2007.

[5] I. Llorente, R. Montero, E. Huedo, K. Leal, A grid infrastructure for utility computing, in: Proceedings of 3rd International Workshop on Emerging Technologies for Next-generation GRID, Manchester, UK, 2006.

[6] R. Wolski, J. Plank, J. Brevik, T. Bryan, G-commerce: Market formulations controlling resource allocation on the computational grid, in: Proceedings of the 15th IEEE International Parallel and Distributed Processing Symposium, San Francisco, USA, 2001.

[7] S. Martello, P. Toth, An algorithm for the generalized assignment problem, Operational Research 81 (1981) 589_603.

[8] E. Munir, J. Li, S. Shi, QoS sufferage heuristic for independent task scheduling in grid, Information Technology Journal 6 (8) (2007) 1166_1170.

[9] S. Kumar, K. Dutta, V. Mookerjee, Maximizing business value by optimal assignment of jobs to resources in grid computing, European Journal of Operational Research 194 (3) (2009) 856_872.

[10] R. Buyya, M. Murshed, D. Abramson, S. Venugopal, Scheduling parameter sweep applications on global grids: A deadline and budget constrained cost-time optimization algorithm, Software Practice and Experience 35 (5) (2005) 491_512.

[11] D. Abramson, R. Buyya, J. Giddy, A computational economy for grid computing and its implementation in the Nimrod-G resource broker, Future Generation Computer Systems 18 (8) (2002) 1061_1074.

[12] R. Wolski, J. Plank, J. Brevik, T. Bryan, G-commerce: Market formulations controlling resource allocation on the computational grid, in: Proceedings of the 15th IEEE International Parallel and Distributed Processing Symposium, San Francisco, USA, 2001.