

Improving the Efficiency of the Task Scheduling Process in Grid System Based Quantum Technology

Ahmad Ali Al-Zubi*

Riyadh Community College, King Saud University, Riyadh 11437, Saudi Arabia.

*Corresponding author. email: ahmadalialzubi01@gmail.com

Manuscript submitted March 13, 2017; accepted May 24, 2017.

doi: 10.17706/jsw.12.6.472-482

Abstract: Quantum technology is one of the emerging field in physics and science engineering which has quantum mechanism properties. This quantum technology has been used in several applications such as quantum imaging, quantum cryptography and quantum computing which are helps to creating the sensing and securing communication. When implementing the sensing communication different environment has been created in which the related task scheduling processing is critical issue while accessing the quantum infrastructure. So, the wide variety of techniques have been proposed in this regard, achieving the efficient scheduling following different schemes. The paper proposes mannerism for improving the performance of GRID-systems through usage of better ways of task scheduling and provides insight regarding improved scheduling algorithm that allows increase in the functional efficiency of Grid systems. Moreover, this study also analyzes the effectiveness of the proposed scheduling method and its comparison with the scheduler Moab Workload Manager baseline methods.

Key words: Quantum technology, grid systems, scheduler, scheduling algorithm, queue, moab workload manager.

1. Introduction

Quantum technology place an important role in the physics and the realted science and technologies. When implementing the quantum functionality based system it requires particular infrastructure, information. Those requirements are accessed from the cloud and grid system via the internet. During the information accessing process, the scheduling of the task is one of the major challenges. At present, the widely adopted Grid system [1] entails the various characteristic properties related to its hardware and software infrastructure [2,3,4] such as Scale of computing resources (memory capacity, number of processors etc) greatly exceeds the resources of a single computer or computing system, Heterogeneity of the environment is made available through computers of different power that are run by different operating systems and the spatial distribution of information and computing resources. Based on the above characteristics, the effective functioning of such systems is heavily dependent on the management of the process computing tasks and resource allocation planning [5,6]. Then the modern grid systems use three main ways of planning [7]: centralized, decentralized and hierarchical [8,9,10]. Currently, there are a large number of task scheduling systems in Grid systems, such as LoadLeveler Multi-Cluster, Condor-G, Oracle

Grid Engine, Open Grid Scheduler, Platform Load Facility Sharing, Moab Grid Suit, Manjarsoft Aneka and also the Moab Workload Manager (MWM).

The basis for each planning system is a tasks processing algorithm, which depicts the selection of the next task from the task list and assigns it a specific host system to perform various functions such as Creating a list of available tasks, prioritizing tasks and creating a queue of tasks in accordance with a given (selected) service discipline, Creating a list of available resources on the basis of the chosen resources allocation method, selecting the first task from the task queue, selecting the first node from the list of resources, appointing a chosen node to the chosen task, shipping tasks to an elected node, If the list of available tasks is not empty - go to no.4 and providing task completion on the basis of the above functions. Even though the algorithm performs various functions, it has some drawbacks such as the search for a resource is always realized since the beginning of the list and selecting the computing node, it does not taken into account the amount of the initial data for the task. So, the planning of each next task, the MWM starts to view the list of resources from the beginning. It leads to the fact that the resource assignment faces uneven behavior because nodes in the beginning of the list will always be more loaded than in the end of list. It also cause a reduction in the rate of scheduling, when N tasks sequential assignment for i-th step, the first i-1 nodes will be already assigned , but the scheduler will still be able to view them. Organization cycle viewing the list of nodes, commencing with the next node after the last assignment node, will allow more uniform loading of the Grid- system. Then the ignoring parameters such as the amount of transmitted data causes inefficiency in the usage of resources. Since the links provided in the system have limited throughput and the data for the task often has a large volume so there is a high probability that the communication channels will be loaded by transferring large amounts of data from one tasks to another as tasks with less volume of data will be located in the queue for transmission. This causes an increase in period of time affecting the start of loading tasks in computing node and the beginning of its implementation. This in turn leads to wastage of computing resources time.

2. Related Work

Qos guided task scheduling technique for the grid computing was proposed in [1]. The Qos guidance was embedded with the general adaptive scheduling heuristics; the algorithm is referred as min-min heuristic and it showed significant performance improvement in terms of the quality of prediction. The approach of Particle Swarm Optimization (PSO) has been employed in [12] for solving the issues of task scheduling in the grid environment. A possible set of solutions is represented for each particle and the continuous variable is transformed into discrete. The optimal schedule is shown to be obtained through this approach which ensures the completion of tasks in the minimum possible time. Also, the PSO algorithm has been compared with the genetic algorithm and the scheduling results were better for the PSO.

An approach for efficient interaction between the resource providers and users have been described in [13] so that the site specific policies can be developed. The interactions of the priority queues and the scheduler, as well as the mechanisms of anti-starvation and fair share are discussed. The job flow and application level scheduling is often done for managing the efficiency of distributed computing environment with the non-dedicated resources [14]. Specific rules are defined for mapping the applications as parallel jobs, and economic scheduling models are used for developing the administration policies.

The schedule based approach has researchers' preference being more influenced by the natural dynamics, in contrast to the traditional queue based approaches. In order to overcome the challenges of dynamically changing grid, event based scheduling approach has been suggested in [15]. The schedules are kept up-to-date according to the events that keep occurring in the grid over time. Scalable grid resource management framework has been developed in [16]. The resources and resource managers within the organization are

arranged hierarchically; and the resources are assigned based on the collaborations of Resource managers. The approach is shown to be more capable than traditional resource managers as these can place workload across the grid more intelligently.

3. The Modified Scheduling

3.1. Algorithm in the Mwm

In order to eliminate drawbacks discussed in the above section, the following modifications can be applied to the basic scheduling algorithm such as Organization cycle viewing the a list of nodes, starting with the next node after the last assignment node, allowing more uniform loading of the Grid- system, Organization of task transmission in computing nodes starting from tasks with less volume of data. It will reduce the waiting time of other tasks in the run queue and The nodes appointment, the connection speed at the lowest and the usage of fewer amounts of data will allow reduction in the time in which tasks are carried out in the run queue. By making the above modification, the system has various benefits such as more uniform loading, reduction in the waiting time of transfer, decreasing the time that the task is in the system and reducing idle of the nodes. Even though the system has various advantages, it has some disadvantages while including the time increment of the task scheduling. So, the following procedure is proposed for the foregoing modified algorithm in terms of determining the list of available task questions, prioritizing the tasks and creating a tasks queue according to the chosen way of tasks prioritization, creating a list of resources according to the selected mode for resource assignments, selecting the group of tasks for planning with the same priority from the tasks queue, sorting a group of tasks for the amount of data that will be processed, selecting a group from the list of resources having same characteristics as the mode selected for resource assignments, sorting a group of nodes for connection speed scheduler, selecting the tasks with the least amount of data from a group, selecting the node with the best connection speed to scheduler, appointing a chosen node to a chosen task, If a group of tasks still have a task, then go to no. 6, if the list of available tasks is not empty, then go to no. 5, sorting of planned tasks for the amount of data, task subscription to the intended nodes and Completion of the work. Then the simulation system, which implements the following algorithms for forming the tasks queuing, is created to analyze the effectiveness of the proposed algorithm in terms of creation of a queue on the FIFO principle, tasks sorting in order of increasing amount of computations, tasks sorting in order of decreasing amount of computations, in this case, the resource assignment was carried out according to the algorithm: first Available and fastest which is shown in the table 1.

Table. 1. Shows the Scheduling Method Analysis Conducted in the Study

No	Scheduling method	Order	
		Forming of tasks queuing	Resource Assignment
1	FIFO/FA	FIFO	First Available(FA).
2	AVC/ FA	Ascending Volume Computation	First Available(FA).
3	RVC/ FA	Reduction Volume Computation	First Available(FA)
4	FIFO/F	FIFO	Fastest (F)
5	AVC/ F	Ascending Volume Computation	Fastest(F)
6	RVC/ F	Reduction Volume Computation	Fastest(F)

4. Analysis of Tasks Execution Time with Different Methods of Tasks Selection and Task Assignment

In this section analyze the effectiveness of the tasks execution time with the different methods for task selection and task assignment process. Fig. 1 shows the dependency ratio of task execution time of base and modified algorithms for the formation of a task queue FIFO and different rules of resource assignment (FA and F). Task execution time in process of planning FIFO / F is less than that in the FIFO / FA.

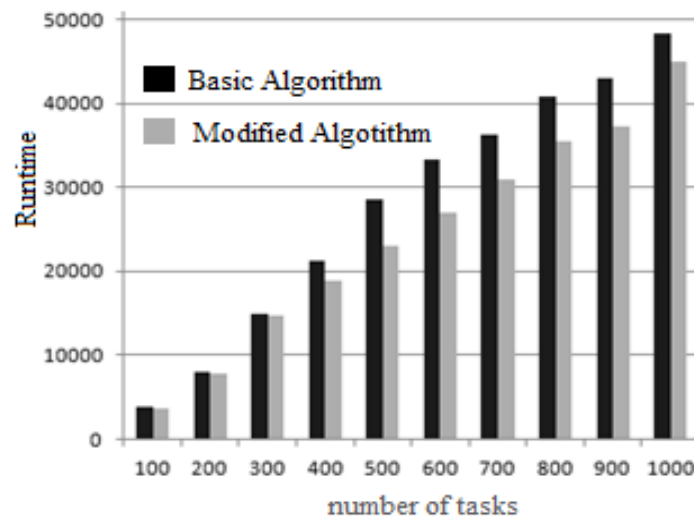


Fig. 1a. FIFO/ FA.

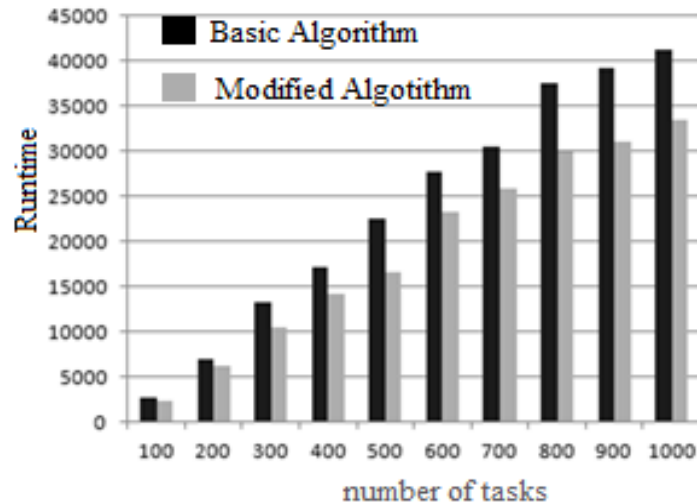


Fig. 1b. FIFO/ F

Fig. 1. Time of task execution using FIFO.

Fig. 1 depicts that the task execution time using a modified algorithm is lesser than that of basic algorithm. At the same time, with increasing number of tasks, the difference in time of execution is increasing. It is explained by the fact that the modified algorithm is characterized by higher data transmission efficiency. The average performance gain when FIFO / FA is 12.4%, while FIFO / F is 22.0% is because the fastest assignment mode activates third modification of the algorithm - assignment of tasks with less data for nodes with low-bandwidth links. Fig. 2 shows the dependency of the task execution time of the base and modified algorithms for formation of a task queue AVC and different rules of resource

assignment (FA and F). Task execution time in a process of planning AVC / F is lesser than that of AVC / FA.

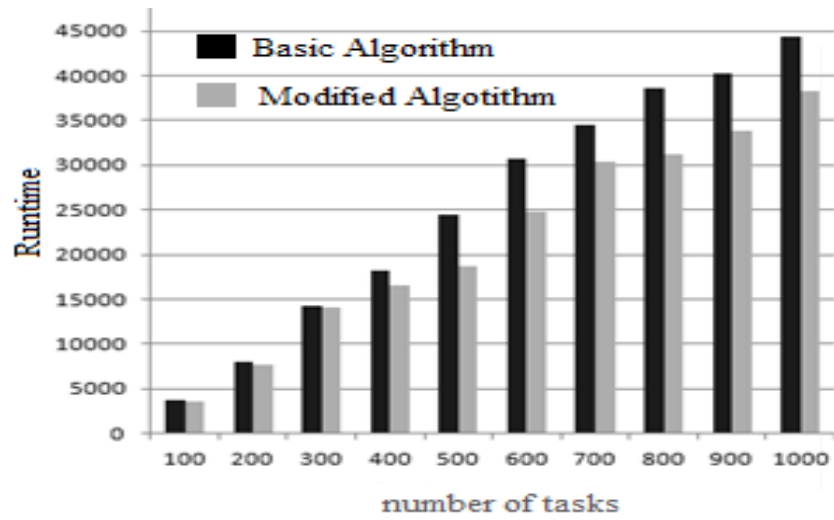


Fig. 2a. AVC/ FA

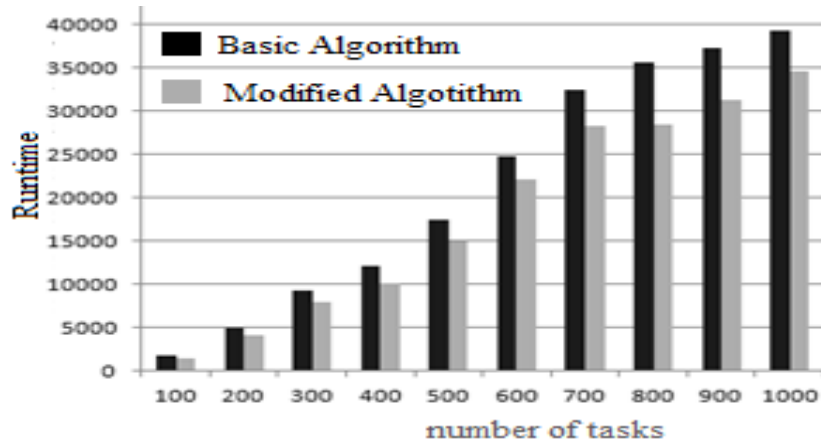
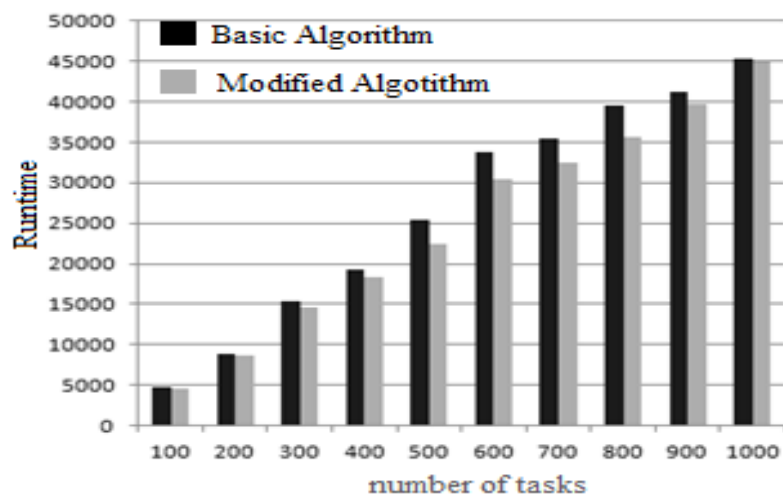
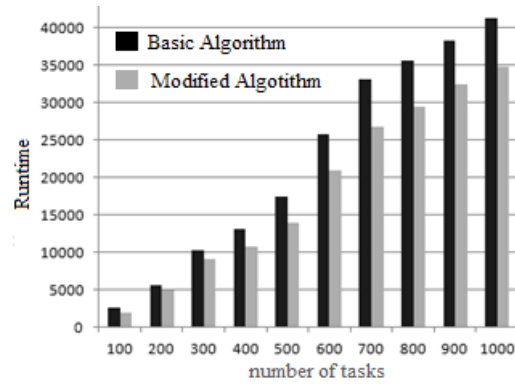


Fig. 2b. AVC/ F

Fig. 2 Task execution time using AVC.

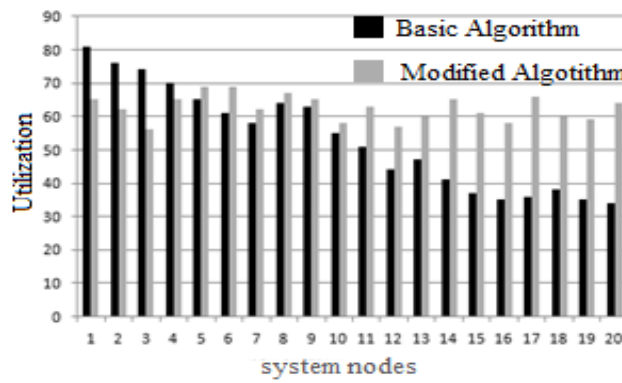


a. RVC/ FA

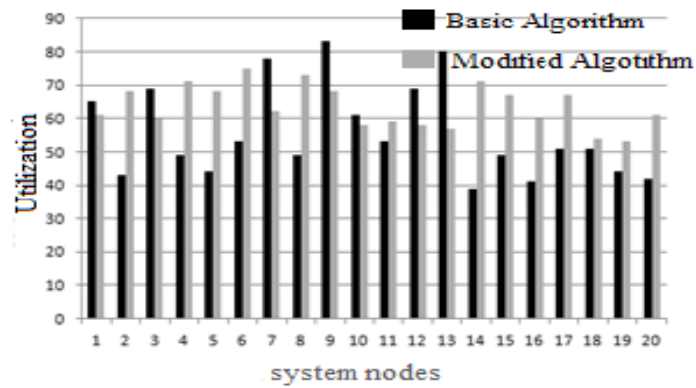


b. RVC/ F

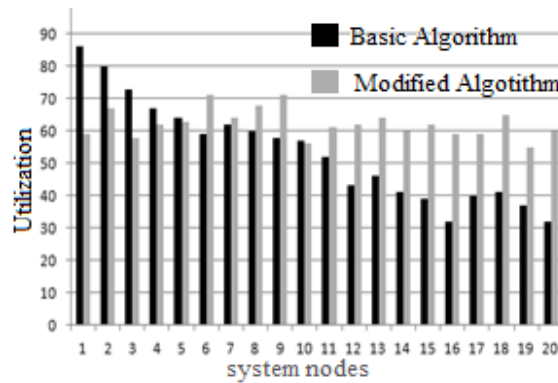
Fig. 3. Task execution time using RVC/ FA method.



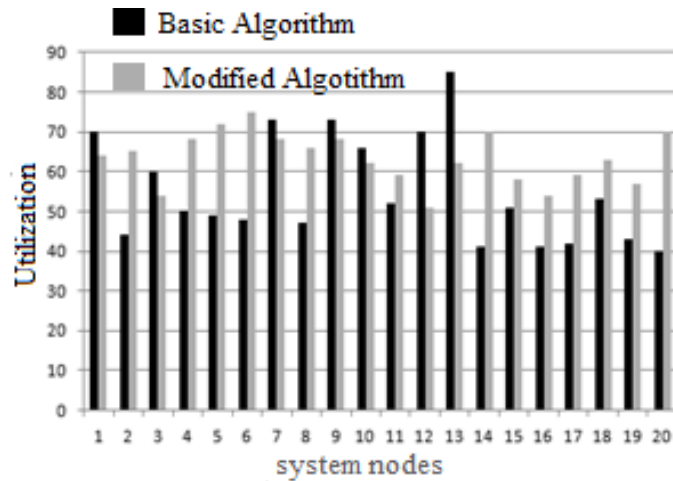
a. FIFO/ FA



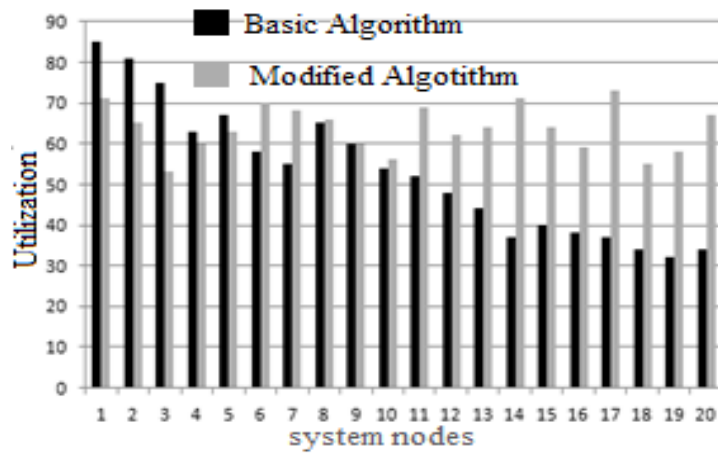
b. FIFO/ F



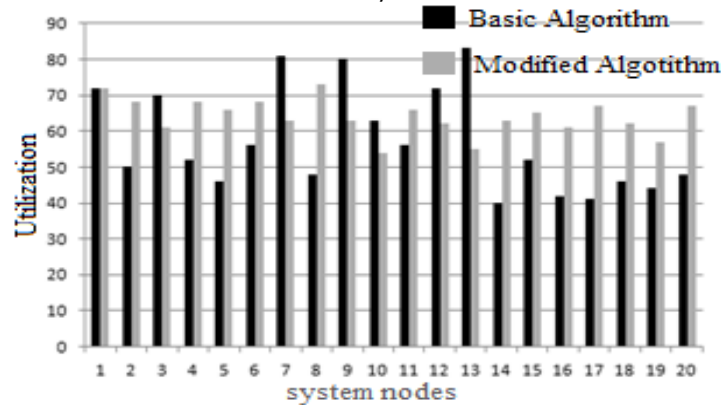
c. AVC/ FAFig



d. AVC/ F



f. RVC/ FA



e. RVC/ F

Fig. 4. Nodes workload using FIFO.

The average growth of productivity in AVC / FA is 14.9%, while in AVC / F, it is 19.9%. Then the Fig. 3 shows the dependency ratio of task execution time of base and modified algorithms for formation of a task queue RVC and different rules of resource assignment (FA and F). Time of task execution in a process of planning RVC / FA is lesser than that of RVC / F.

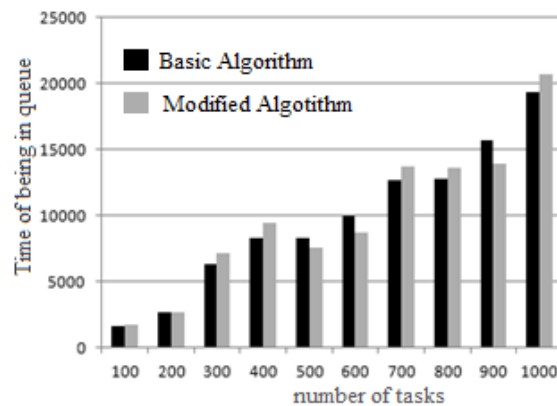
Average increase of productivity in RVC / FA is 6.5%, while in RVC / F, it is 19.9%. As in the previous cases, the task execution time using the modified algorithm is lesser than in case of using basic algorithm.

But in this case the difference between the values of various algorithms is significantly large.

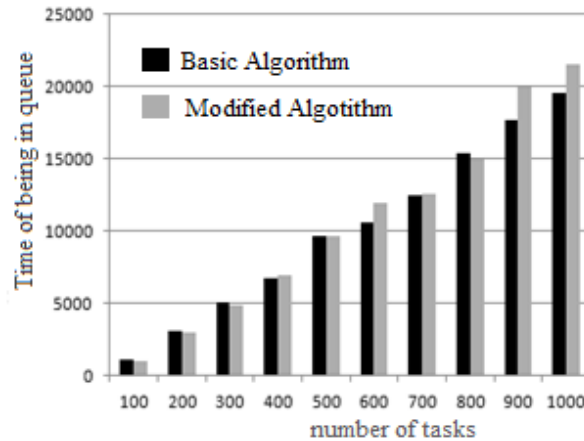
5. Analysis of the Nodes Workload in Grid System at Different Methods of Task Processing Organization

In this section analyze the nodes workload in grid system at different methods of task processing organization. The Fig. 4 depicts the dependency of loading nodes of basic and modified algorithms for the formation of a task queue FIFO and different rules of resource assignment (FA and F).

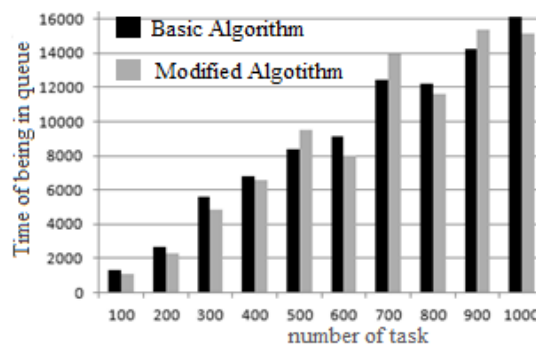
From Fig 4, we can observe that the use of modified algorithm in all cases gives more uniform loading of the nodes. The histogram illustrates that in case of basic algorithm and method for allocating resources, FA first nodes in the list is the most downloaded one. This is due to the fact that the scheduler assigns tasks to the nodes according to their sequence number.



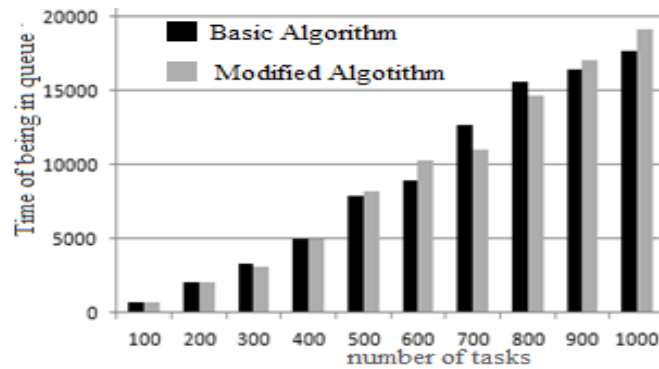
a. FIFO/ FA



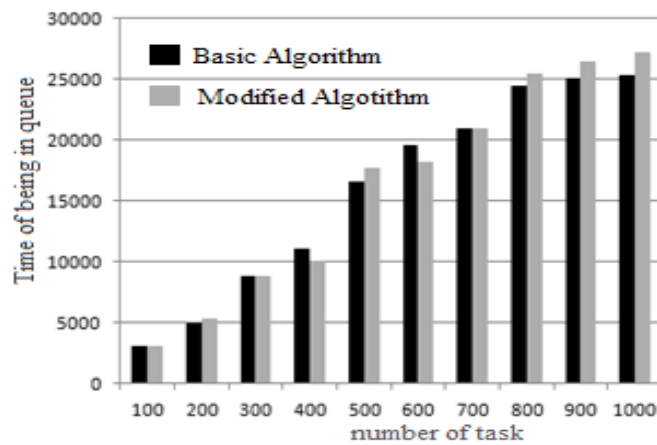
b. FIFO/ F



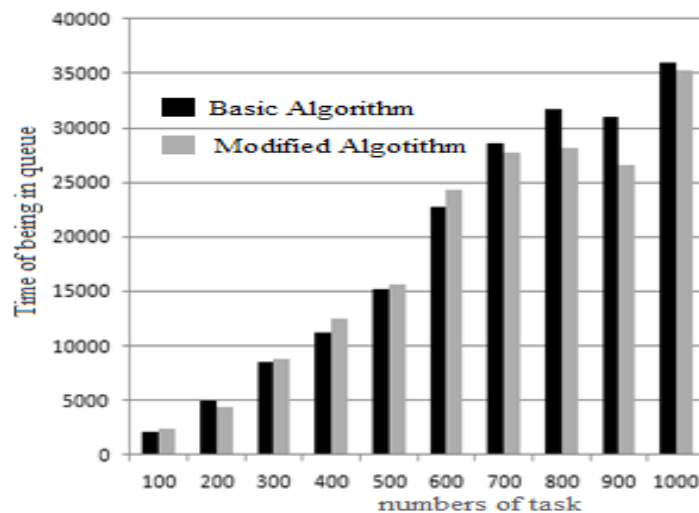
c. AVC/ FA



d. AVC/ F



f. RVC/ FA



e. RVC/ F.

Fig. 5. The task residence time in the queue.

6. Analysis of the Residence Time of the Tasks in A Queue

Fig. 5 shows a comparative histogram task residence time in the scheduler queue depending on the number of tasks in various means of processing. From the histogram, we can see that there are no significant changes in the task residence time in the scheduler queue. This is due to the fact that the scheduler modification does not affect the process of tasks queuing formation.

The histogram shown in Fig. 5. illustrates that when the process of appointment of F (Fastest) tasks residence time in a queue is more than that of FA, then the biggest time of finding tasks is in queue RVC / F due to inefficient use of transmission channels.

7. Conclusions

As the resulted data shows that using a modified algorithm facilitates in performance increase in Grid-systems in comparison to using a base algorithm in the range of 6.6% to 22%. For the Fastest assignment mode, the First Available increase is smaller than the fastest mode. It is due to the fact that the third modification of the algorithm (destination of nodes, the connection speed as the lowest with tasks having less data) is not intended to the First Available mode. The result of the workload of nodes using basic algorithm is always irregular. For the mode of First Available node, the workload depends on the serial number as presented in the scheduler list _ - first nodes are most downloaded. For the fastest mode, more loaded ones are considered as the most productive nodes. Using a modified algorithm nodes ensures workload uniformly for all work modes. The residence time of tasks is least in the scheduler queue when using the sorting of tasks for weight calculation. In case of increase, a little more time is availed when creating a queue in FIFO and it is the highest when task sorting is based on churning weight calculations. In general, the use of a modified algorithm does not affect the residence time of tasks in the scheduler queue.

Furthermore, improvement in the utilization of computing resources of Grid-system is the advanced computational device load before the end of previous calculations. The presence of each compute node input tasks of loading buffer will reduce the downtime of computing node.

References

- [1] Foster, C. K. (1998). The grid: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, San Francisco, Calif, USA.
- [2] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., & Tuecke, S. (2001). The data GRID: Towards an architecture for the distributed management and analysis of large scientific data sets. *J. Network and Computer Applications*.
- [3] Childers, L., Disz, T., Olson, R., Papka, M. E., Stevens, R., & Udeshi, T. (2000). Access GRID: Immersive group-to-group collaborative visualization. *Proceedings of the 4th International Immersive Projection Technology Workshop*.
- [4] Czajkowski, K., Fitzgerald, S., Foster, I., & Kesselman, C. (2001). GRID information services for distributed resource sharing.
- [5] Dong, F. (2000). Scheduling algorithms for grid computing: State of the art and open problems. *Technical Report No. 2006-504 School of Computing*, Queen's University Kingston, Canada.
- [6] MohdKamir, Y. (2010). Comparative study a performance and capability scheduling techniques in grid computing. *International Journal on Computer Science and Engineering*.
- [7] Hamscher, V., Schwiegelshohn, U., Streit, A., & Yahyapour, R. (2000). Evaluation of job-scheduling strategies for grid computing. *Proceedings of the 4th GRID 2000 GRID 2000, First IEEE/ACM International Workshop*.
- [8] Chauhan, P. N. (2012). Resource based optimized decentralized grid scheduling algorithm. *Advances in Computer Science, Engineering and Applications*.
- [9] Chauhan, P. N. (2012). Fault tolerant decentralized scheduling algorithm for P2P grid. *Proceedings of the 2nd International Conference on Communication, Computing and Security*.
- [10] Dong, Z., Yang, Y., Zhao, C., Guo, W., & Li, L. Computing field scheduling: A fully decentralized scheduling approach for grid computing. *Proceedings of the 6th Annual ChinaGrid Conference* (pp. 68–73).

- [11] Dong, Z., Yang, Y., Zhao, C., Guo, W., & Li, L. (2011). Computing field scheduling: A fully decentralized scheduling approach for grid computing. *Proceedings of the 6th Annual China Grid Conference*.
- [12] Zhang, L., Chen, Y., Sun, R., Jing, S. & Yang, B. (2008). A task scheduling algorithm based on PSO for grid computing. *International Journal of Computational Intelligence Research*, 4(1), 37-43.
- [13] Klusáček, D., & Tóth, S. (2014). On interactions among scheduling policies: Finding efficient queue setup using high-resolution simulations.
- [14] Toporkov, T. V., Tselishchev, A., Yemelyanov, D., & Bobchenkov, A. (2012). Composite scheduling strategies in distributed computing with non-dedicated resources, *Procedia Computer Science*.
- [15] Klusacek, D. (2008). Dealing with uncertainties in grids through the event-based scheduling approach, *Fourth Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*.
- [16] Mohamed, E., & Krishnamurthy, D. (2006). A scalable wide-area grid resource management framework. *Proceedings of the International Conference on Networking and Services*.



Ahmad Ali Al-Zubi is with the Riyadh Community College, King Saud University, Riyadh 11437, Saudi Arabia.