Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.019928
Article

Tech Science Press

Scheduling Algorithm for Grid Computing Using Shortest Job First with Time Quantum

Raham Hashim Yosuf[1], Rania A. Mokhtar[2], Rashid A. Saeed[2,*], Hesham Alhumyani[2] and S. Abdel-Khalek[3]

[1]Electronics Engineering Department, Sudan University of Science and Technology, Khartoum, 1111, Sudan
[2]Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia
[3]Department of Mathematics, College of Science, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia
[*]Corresponding Author: Rashid A. Saeed. Email: abdulhaleem@tu.edu.sa

Received: 02 May 2021; Accepted: 05 June 2021

Abstract: The grid computing is one of the strong initiatives and technologies that has been introduced in the last decade for improve the resources utilization, optimization and provide very high throughput computation for wide range of applications. To attain these goals an effective scheduling for grid systems is a vital issue to realize the intended performance. The processes scheduling could be executed in various methods and protocols that have been extensively address in the literature. This works utilized shortest process first (SPF) protocol which gives the shortest jobs the highest priorities. For longer jobs, it should have lower priorities and wait in the queue longer time. With too many smallest tasks reach the queue, the long tasks can't have an opportunity to be processed and the number of tasks continue growing in front of them, which is well-known issue called Starvations. In this work we introduce a new time quantum-based for shortest process first to avoid the starvation problem. The objective of the model is to optimize the utilization of resources by reducing end time of the job and enhancing performance of job scheduling. The model introduces fair treatment for all the processes. A simulation environment for the proposed model is developed and implemented. The results show that the quantum time-based solution has improved the starvation issues severely in terms of time delay, optimization of grid resources and fairness. The results show that SJF with time quantum outperform the SJF in terms of end time process, resources utilization, delay to transmission and Fair treatment for all the processes. But in term of waiting time Shortest Job First is better, SJF does not always minimize waiting time.

Keywords: Grid computing; scheduling; time quantum shortest process first

1 Introduction

A Grid computing is state-of-the-art technology that recently has been grown very fast in industry and academia. Recently, many new sophisticated and complex applications with high resources and computation need i.e., nanotechnologies, machine learning, nuclear energy etc. [1]. Grid systems are technique and mechanism for computation resources sharing and utilization in a distribution manner that can provide highly efficient services for distributed and parallel systems that allows the share, select, aggregate and distribute computation resources for several organizational zones based on their capability, availability, cost, performance and users' quality of services (QoSs) required [2]. Resources could be memories, processors, applications, buses, and information all are structured and organized in a such away to be associated to Internet connection and a firmware layer. This structure for grid computing to offer elementary facilities for privacy, supervision, and administration for resources. Resources maintained by several organizational administration are pooled using local declared strategies and policies. Policies are set for organizing the what/who/how the resources to be shared and distributed among the clients/applications. The grid computing is one of the strong initiatives and technologies that has been introduced in the last decade for improve the resources utilization, optimization and provide very high throughput computation for wide range of applications. To attain these goals an effective scheduling for grid systems is a vital issue to realize the intended performance. The processes scheduling could be executed in various methods and protocols that have been extensively address in the literature.

The typical scenario for the current utilization of resources in many organizations, is that there are huge number of resources are been underutilized and wasted due to duplication and. Grid computing systems afford a platform for employing and optimizing the usage of resources and then have the opportunity of significantly improve resources utilization efficiently. Grid computing looks like a great answer to numerous problems without prodigious management requirements [3]. Effective grid computing is possible, however, only if the resources are scheduled well, the main aim of resources scheduling is the resources utilization enhancement and reduce the tasks time processing.

The task schedule protocol and procedure function are to elect the best appropriate a grid processor/resource for the clients'/users' tasks. Schedule arbiter allocates tasks in the available resources for execution utilizing various algorithms such as First Come First Service (FCFS), Longest Job First (LJF), Earliest Deadline First (EDF) and other algorithms. One type of these algorithms is shortest jobs first (SJFs) also identified as Shortest Jobs Next (SJNs) or Shortest Processes Next (SPNs) which are the mechanism of queue schedule for selecting the shortest jobs [4].

In this work shortest jobs first (SJFs) method is used in which the shortest jobs are being assigned with higher priorities. If jobs and tasks are lengthy, these tasks would suffer from high delay in the queue. With too many smallest tasks reach the queue, the long tasks can't have an opportunity to be processed and the number of tasks continue growing in front of them, which is well-known issue called Starvations. The objective of the model is to optimize the utilization of resources by reducing end time of the job and enhancing performance of job scheduling. The model introduces fair treatment for all the processes. A simulation environment for the proposed model is developed and implemented. The results show that the quantum time-based solution has improved the starvation issues severely in terms of time delay, optimization of grid resources and fairness. The results show that SJF with time quantum outperform the SJF in terms of end time process, resources utilization, delay to transmission and Fair treatment for all the processes. But in term of waiting time Shortest Job First is better, SJF does not always minimize waiting time.

The reminder of the paper has been arranged and structured as: section 1 discussed the motivation of the topic and the problem statement. Section 2 presents the related works and literature. The proposed algorithm-based time quantum is proposed in section 3. In section 4 presents the achieved results and the simulation scenarios environments. Lastly, a conclusion for the paper is given.

2 Related Works

The task schedule protocol and procedure function are to elect the best appropriate a grid processor/ resource for the clients'/users' tasks. Schedule arbiter allocates tasks in the available resources for execution utilizing various algorithms. This section describes the algorithms and determines the difference between them.

In Sharma et al. [5] the challenge of task schedule in queue for grid computing resources is discussed. The paper presents a few scheduling methods for grid applications and services. A benchmarking between various techniques have been presented and analyzed. The the paper proposed a new schedule technique called Earlier deadlines first with shortest remain times (EDSRTFs) where schedule processes and procedures are executed conferring to the tasks' deadlines and remain times for tasks. The jobs with the earlier deadlines would be able to allocate resources first whereas the jobs with larger deadlines value would have to delay regardless of its processing times. by Comparison of the proposed algorithm EDSRTF in relations of delay and reversal jobs time with another two techniques first come first served (FCFSs) and longer jobs first (LJFs) [6]. Mean time delay and mean time reverse for the proposed mechanism has been found shorter when compared to other schedule mechanisms; so EDSRTFs was found better than FCFSs and LJFs techniques.

In Jain et al. [7], the main aim is to implement a new mechanism for SJFs schedule protocol that assist to improve the starvations challenge and problematic issue in a severely loaded computing systems are discussed. The A adaptive SJFs procedure is proposed to enhances the starvations problematic issue that the conventional and basic SJFs structure suffers from it. This adaptive SJFs approach is relies on eight SJFs and multi-level queue feedback schedules (MQFS) algorithm. The paper develops a comparison study between adaptive SJFs and SJFs algorithms. The authors determined that adaptive SJFs method is outperform the SJFs in term of starvation degree minimization of, fairness enhancement, reduction of time response and less time of resources assignments to each task and job. Adaptive SJFs approach is shows that clearly maximize the utilization of CPU and resources efficiency. As MQFS approach is combined with SJFs, the resulted mechanism is processes divide for several queues and processes switching amongst themselves would greatly enhance starvations problem.

In Kumaresh et al. [8], the authors present an approach that only preserve the utmost requested data/ information blocks in cache memory. The preserve is depending on the reuse frequency of data blocks. The proposal is identifying the data rate of usage using statistical approach. The enhancement of the proposed scheme is by using caching mechanism that permits the grid to decrease the cost of computational by using the previous calculations that stored in the cache without repeating the [processing task. Fig. 1 shows the scheduling algorithm structures.

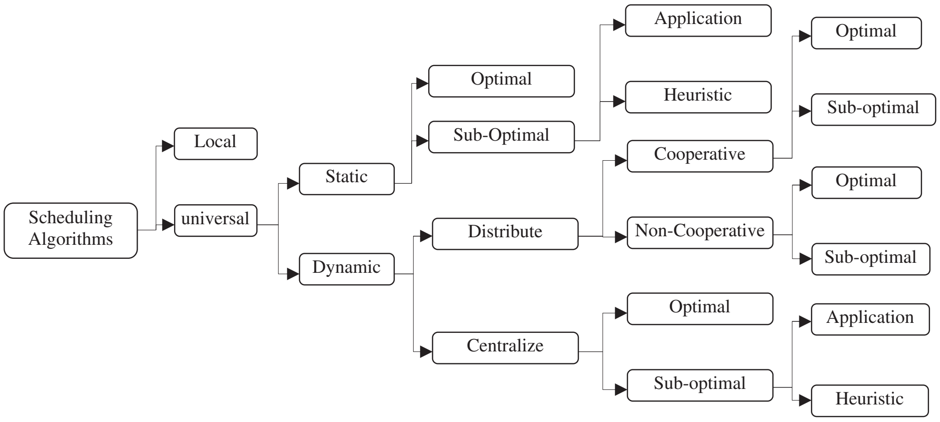

Figure 1: Scheduling algorithm structures

In Khaliq et al. [9], the proposed schedule mechanism which categorizes the sub-processes on priority based. Multi-levels queues are utilized for starvation enhancement. The invented zero-starvation (ZS) Schedule mechanism could be implemented in wide range of applications with efficient resources schedule, especially in sensitive grid applications to starvation. The achieved simulation results prove that the proposed scheme have attained resources optimization with high rate of utilization [10]. The multi-levels queues-based schedule scheme for the heterogeneous grid computing structure have excellent throughput when benchmarked to the other related schedule mechanisms under similar environment.

In Awan et al. [11], the paper presents a task scheduling solution by utilizing Improved Multi-Levels Queues based deadline (MMLQ-D) for multiprocessors schedule. The invented mechanism combined two important algorithms for managing tasks assignments in the queues for execution through MMLQs protocol. The method presents the issue of low priorities processes starvations, hence enhancing the multiprocessors grid systems capability. MMLQ-D mechanism has been benchmarked with earliest deadlines first (EDFs) scheme, where MMLQ-D shown more resources utilizations and outperform EDF.

In Eisa [12], the author proposed two separate features and mechanisms: external schedule and internal schedule. The paper assumes a design with a combined external and internal queue scheduling for both global and local perspectives. The design emphases on utilization of resources enhancement at global queues level and reduce waiting and delay times for local queues level. The main aim of the proposed mechanism to explore the grid computing schedule and queue problem issues to discuss the hybrid mechanism for global/local queues features and functionalities.

In Rosemarry et al. [13] a mechanism is modeled for addressing an effective task queuing systems with jobs scheduling mechanisms to improve the utilization of grid resources and reduce the processes delay and waiting times in the queue. Authors have discussed three queuing algorithms (SJFs, FCFSs, Round Robin (RR) for tasks schedule and queue management. The paper has achieved excellent processing waiting time and delay at various simulation scenarios and queue policies [14]. Also, the paper has developed SJFs scheme as elementary queuing procedure with the proposed policies and scheduling algorithm.

In Rosemarry et al. [15], the authors proposed an enhanced approach based on tasks priorities for queue schedule where the performance parameters were tasks waiting time in queue and utilization rate for the grid available resources. The approach was introducing new clustering mechanism where ten priorities levels were identified, then, the new proposed scheduling is compared with other four conventional algorithms i. e., SJFs, FCFSs, RR and priority-based Scheduling. The performances were illustrated and discussed and critical discussion for various consideration in the proposed simulation model.

In [16] they are proposing a backfilling scheduling algorithm is proposed to select many jobs to be backfilled from the waiting job queue. The Main aim of their proposed work is to make develop scheduling jobs based on priority using backfilling for grid computing. In [17] they are proposing a backfilling scheduling algorithm is proposed to select many jobs to be backfilled from the waiting job queue. The Main aim of their proposed work is to make develop scheduling jobs based on priority using backfilling for grid computing.

3 Proposed Model

As shown in Fig. 2, SJF algorithm the resource is allocated to the smallest process. If process is long, then this long process is suffering with a long waiting time. If short Processes are always available to run, the long processes ones never have opportunity to be scheduled. This challenge is known as starvations. In this paper a proposed solution is to integrate the shortest job first with time quantum, the aim is to develop an approach that employ SJF scheduling algorithm by using time quantum to reduce the starvation problem. The model relay on the integration of round robin (RR) and SJF scheduling algorithm.

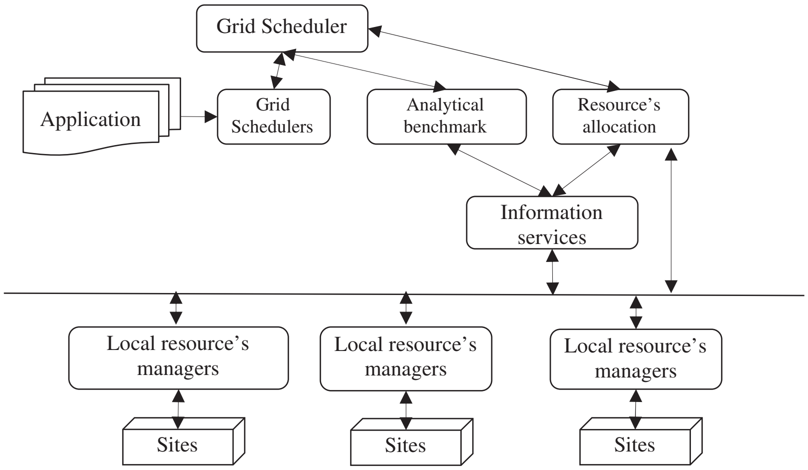

Figure 2: The grid scheduler model

In this algorithm the resource is allocated to the shortest process. In this scenario two things can happen. First, the process may be less than or equal to the time quantum. In this case, the process will execute and after completion release the resource by itself. Second, the process may be greater than to the time quantum. In this case, the process will execute for one time quantum, divided and distributed to the free second resource, if resources are busy the process is preempted. Then, the resource scheduler will select the next shortest process to execute. The preempted process will be put at the ready queue. This continues until the execution of all the processes is completed. The objective of the model is to optimize the utilization of resources by reducing end time of the job and enhancing performance of job scheduling. The model introduces fair treatment for all the processes.

The idea is to enhance SJF algorithm using time quantum. In this model the channel is allocated to the shortest process, where two possible things can happen, first, if process less than or equal to the time quantum, then the process will be executed and after completion release the channel by itself [18]. Second. If process greater than time quantum, then the process will execute for 1 time quantum, divided, and distributed to the free second channel, if channels are busy the process is preempted. Then, the channel scheduler will select the next shortest Process to execute. The preempted process will be put at the ready queue. This continues until the execution of all the processes is completed.

4 Simulation Results

4.1 SJF Algorithm

First consider the SJF algorithm without time quantum, for a scenario of data rate value equal 60, three channels (channel 1, channel 2, channel 3) and the process is not arranged. After arrange data all channels are free that means select minimum processes (30 60 70) to allocated channel 1, channel 2 and channel 3 [19]. Channel 3 is become free because it is having minimum data and process 80 is allocated in it. When process complete execution it releases channel by itself. That means process 400 is allocated in channel 3, processes 100 and 500 is allocated in channel 2 and process 200 is allocated in channel 1. Tab. 1 shows SJF scheduling algorithm. Fig. 3 shows the performance of SJF without time quantum.

4.2 SJF Algorithm-Based Time Quantum

Considering a scenario of data rate value equal 60, time quantum equal 180, the channels are 20 (channel 1, channel 2, channel 3) and the process is not arranged. After arrange data all channels are free that means select minimum processes (30 60 70) to allocated channel 1, channel 2 and channel 3 [20].

Algorithm 1:

```
Select channel.
Select shortest process from the ready queue.
Check, if process <= Time quantum (T.Q)
If yes, then scheduler will allocate the channel to that process, the process will complete its execution and will release the channel by itself.
If no, then scheduler will allocate the channel to the process, the process will execute for one T.Q, divided and distributed to the free second channel
If channels are busy the process is pre-empted.
Repeat step 3, if any process is available in ready queue.
Exit
```

Table 1: Shortest job first (SJF)

Processes	Waiting time	End time	Delay to free
30	0	0.5000	0.5000
60	0	1	1
70	0	1.1667	1.1667
80	0.5000	1.833	1.333
100	1	2.667	1.667
200	1.1667	4.5000	3.333
400	1.833	8.5000	6.667
500	2.667	11	8.333

Channel 1 is became free because it is have minimum data and process 80 is allocated in it and process 100 is allocated into channel 2 after process 80 complete its execution and release channel process 200 is greater than time quantum (180) the process is divided to 180 and 20 and distributed into channel 1 and channel 3, process 400 is also greater than time quantum divided to 180, 180, 40 and distributed into channel 1, channel 2 and channel 3, process 500 is greater than time quantum divided to 180,180,140 distributed into channel 1, channel 3 and channel 2.

End time of all processes that is the time when process is complete its execution and release channel. For example, process 30 is start its execution at 0 and finish at 0.5000 second that means end time of execution and delay to free calculate using the difference between the waiting time and end time. Tab. 2 shows SJF scheduling algorithm with time quantum. Fig. 4 shows the SJF algorithm-based time quantum performance for different number of processes.

4.3 SJF Algorithm-Based Time Quantum with Different Data Rates

Now considering a scenario when data rate is different (20 80 100). After arrange data all channels are free that means select minimum processes (30 60 70) to allocated channel 1, channel 2 and channel 3. All processes are greater than time quantum must be divided and distributed into free channels, when process is complete execution, it releases channel by itself. Tab. 3 shows SJF scheduling algorithm-based time

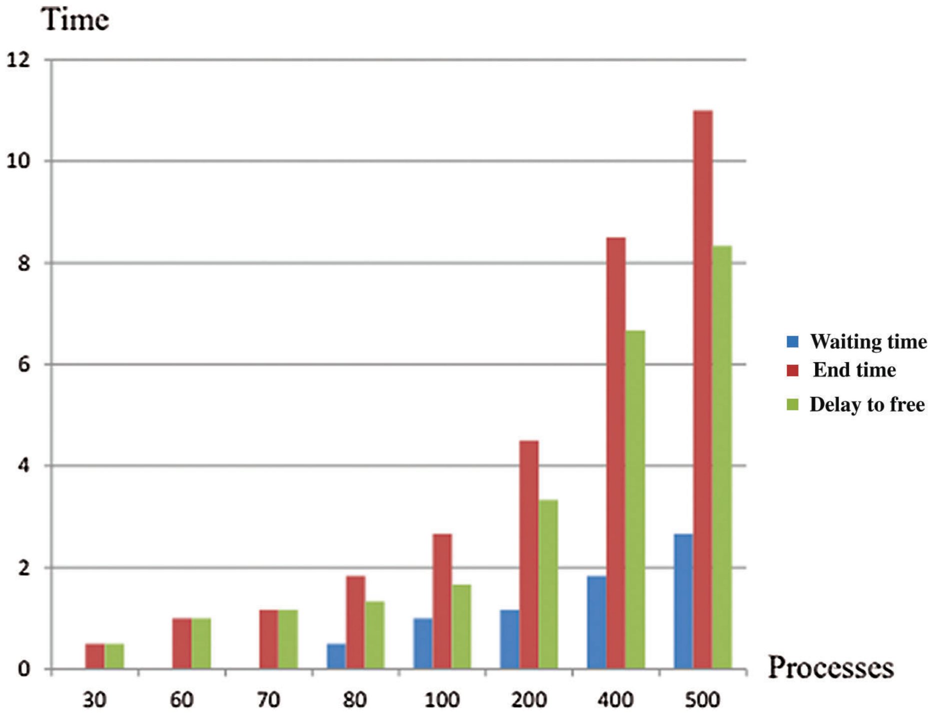

Figure 3: The performance of SJF without time quantum

Table 2: Shortest job first algorithm with time quantum

Processes	Waiting time	End time	Delay to free
30	0	0.5000	0.5000
60	0	1	1
70	0	1.167	1.167
80	0.5000	1.833	1.333
100	1	2.667	1.667
200	1.167	4.167	3
400	2.167	5.667	3.5
500	4.833	8.167	3.33

quantum with different data rates. Fig. 5 shows the illustration of various data rate for SFJ-based time quantum under different number of processes.

The result shown in Figs. 3–5, presents that SJF with time quantum outperform the SJF in terms of end time process, resources utilization, delay to transmission and Fair treatment for all the processes. But in term of waiting time Shortest Job First is better, SJF does not always minimize waiting time.

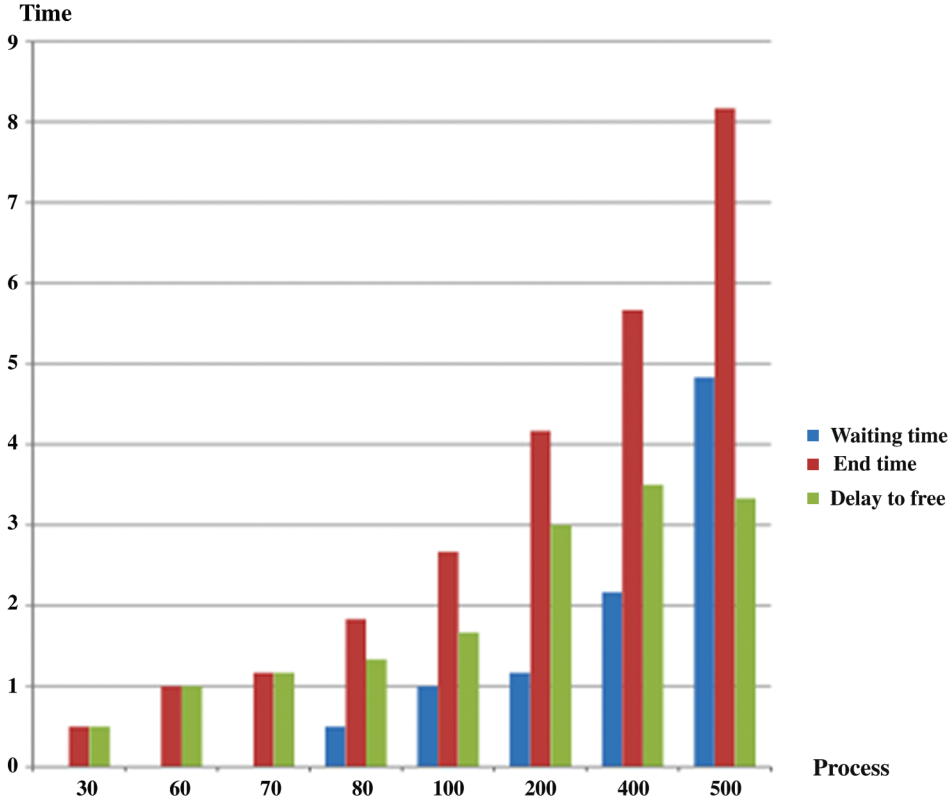

Figure 4: Shortest job first algorithm-based time quantum for three parameters (waiting time, end time and delay to free)

Table 3: Various data rates for SJF-based time quantum

Processes	Waiting time	End time	Delay to free
30	0	1.5000	1.5000
60	0	0.7500	0.7500
70	0	0.7000	0.7000
80	0.7000	1.5000	0.8
100	0.7500	2	1.25
200	1.5000	10.5000	9
400	1.7000	4.2500	2.55
500	3.9000	7.1000	3.2

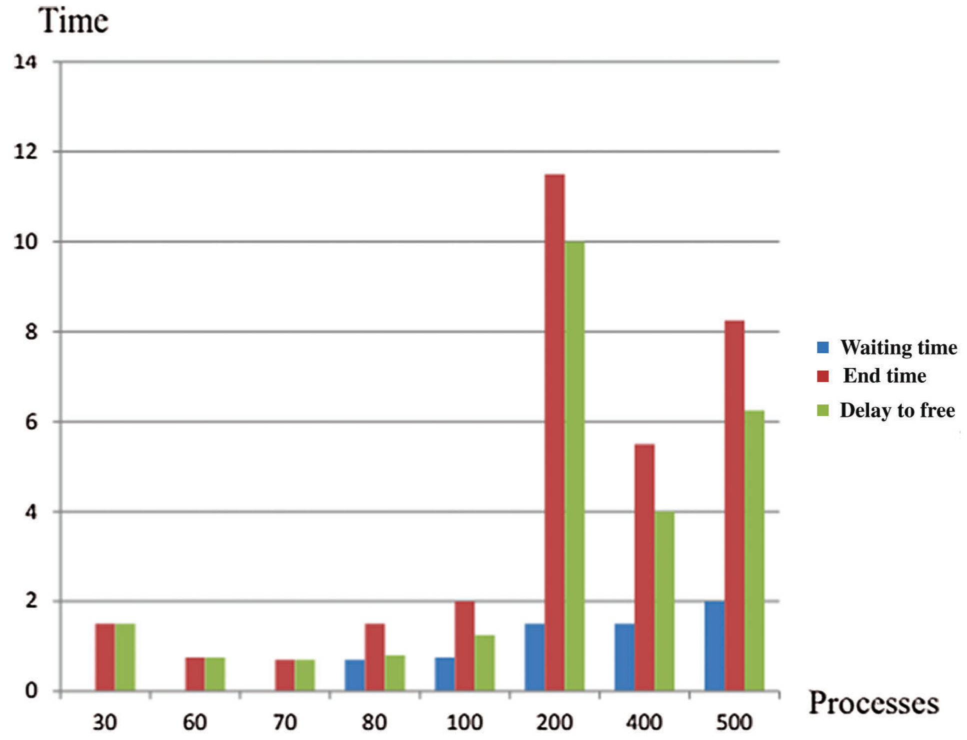

Figure 5: Shortest job first algorithm-based time quantum with various data rates

5 Conclusions

Grid computing could be a solution for more sophisticated processes in short time and help in efficiently resources utilization. For grid to works appropriately, efficient jobs schedule policies and strategies should be implemented. Task schedule assists the tasks to have resources allocated accurately. This paper proposed a new schedule technique that combined shortest task first with time quantum and also a comparison has been conducted in term of task time waiting in queue, end time and delay to transmission of jobs with algorithm Shortest Job First (SJF). Average waiting time, average end time and average delay to transmission of jobs are also calculated. The proposed model achieves better resources utilization and fairness among different processes, it has several benefits in comparison to SJF algorithm in term of Low-end time, Low delay to transmission, and minimal degree of starvation. In future Shortest Job First with adaptive quantum time and would be also utilizing adaptive quantum time for prepared queues, where quantum time would be computed each time, a task allowed to enter or exit the queue. The proposed mechanism is outperforming the other conventional methods in term of the grid throughput, performance and reduce the delay time.

Funding Statement: This Research was supported by Taif University Researchers Supporting Project Number (TURSP-2020/216), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. A. Saeed, S. A. Al-Talib, T. A. Al Ahdal, H. Mohamad, M. Abbas *et al.,* "Adaptive RS-group scheduling for WiMAX multihop relay," in *Proc. ICCCE*, KL, Malaysia, pp. 550–554, 2010.

[2] R. V. Sumathy, "A detailed study of resource scheduling and fault tolerance in grid," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 357–361, 2011.

[3] N. A. Azeez1, A. P. Idoye, A. O. Adesina, K. K. Agbele, I. Tiko *et al.,* "Peer to peer computing and grid computing: Towards a better understanding," *Pacific Journal of Science and Technology*, vol. 12, no. 1, pp. 270–276, 2011.

[4] M. Bhardwaj and S. Kumar, "A two-way scheduling approach for effective resource scheduling in grid," *International Journal of Computer Networks and Wireless Communications*, vol. 3, no. 3, pp. 243–246, 2013.

[5] D. Sharma and P. Mitta, "Job scheduling algorithm for computational grid in grid computing environment," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, pp. 735–743, 2013.

[6] H. Aggarwal and S. Nagpal, "Augmented SJF algorithm with reduced starvation," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, pp. 718–723, 2014.

[7] H. Jain and K. Khatkar, "Process scheduling approach for starvation improvement with time delay analysis in grid resources allocation," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 8, pp. 197–201, 2015.

[8] V. S. Kumaresh, S. Prasidh, B. Arjunan, S. Subbhaash and M. K. Sandhya, "Multilevel queue-based scheduling for heterogeneous grid environment," *International Journal of Computer Science*, vol. 9, no. 3, pp. 245–248, 2012.

[9] S. Khaliq, T. Maqsood, M. Ali, K. Bilal, S. A. Madani *et al.,* "A load balanced task scheduling heuristic for large-scale computing systems," *Computer Systems Science and Engineering*, vol. 34, no. 2, pp. 79–90, 2019.

[10] M. Hasija, A. Kaushik, S. Kaushik and M. Barnela, "D-MMLQ algorithm for multi-level queue scheduling," *International Journal of Computer Science and Network Security*, vol. 14, no. 7, pp. 90–94, 2014.

[11] N. Awan, S. Khan, M. Khalid, M. Tahir, N. Alam MD *et al.,* "Machine learning-enabled power scheduling in IoT-based smart cities," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2449–2462, 2021.

[12] M. Eisa, "Improving grid computing scheduling using heuristic algorithms," *International Journal of Computer Applications*, vol. 78, no. 6, pp. 14–17, 2013.

[13] P. Rosemarry, P. Singhal and R. Singh, "A study of various job & resource scheduling algorithms in grid computing," *International Journal of Computer Science and Information Technologies*, vol. 3, no. 6, pp. 5504–5507, 2012.

[14] V. Pandey and P. Saini, "Application layer scheduling in cloud: Fundamentals, review and research directions," *Computer Systems Science and Engineering*, vol. 34, no. 6, pp. 357–376, 2019.

[15] P. Rosemarry, R. Singh, P. Singhal and D. Sisodia, "Grouping-based job scheduling algorithm using priority queue and hybrid algorithm in grid computing," *International Journal of Grid Computing & Applications*, vol. 3, no. 4, pp. 55–65, 2012.

[16] F. Alsolami, F. A. Alqurashi, M. K. Hasan, R. A. Saeed, S. Abdel-Khalek *et al.,* "Development of self-synchronized drones' network using cluster-based swarm intelligence approach," *IEEE Access*, vol. 9, pp. 48010–48022, 2021.

[17] Q. Mateen, U. Niazi and S. Marwah, "Grouping based job scheduling algorithm using priority queue, shortest job first, round robin and first come first serve," *International Journal of Computer and Communication System Engineering*, vol. 2, no. 1, pp. 139–142, 2015.

[18] L. Rui1, Y. Qin, B. Li and Z. Gao, "Context-based intelligent scheduling and knowledge push algorithms for AR-assist communication network maintenance," *Computer Modeling in Engineering & Sciences*, vol. 118, no. 2, pp. 291–315, 2019.

[19] S. F. Lokhande, D. C. Sachin and S. R. Jadhao, "Grid computing scheduling jobs based on priority using backfilling," *International Journal of Electrical Electronics & Computer Science Engineering*, vol. 2, no. 2, pp. 68–72, 2015.

[20] E. Nabiel, S. P. Lee, S. U. Khan, N. Behboodian, O. Ibrahim *et al.,* "A heuristics-based cost model for scientific workflow scheduling in cloud," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3265–3282, 2021.