

# Tasks Scheduling in Private Cloud Based on Levels of Users

Dr. Mohammed A. Fadhil Al-Husainy

**Abstract—** In private cloud computing, tasks scheduling algorithms play a major role in the performance of the cloud system and the Quality of Services (QoS) that is provided to users. But, most current algorithms usually neglect the existence of different types / levels of users who working in the same institution. Tasks scheduling algorithm named Best-Level-Job-First (BLJF) has been proposed in this paper which takes into consideration the levels of users in the institution, in addition to the other commonly used parameters in scheduling tasks, when arranging the tasks of users in the queue for execution by the private cloud. The user level in the institution has been added as a new parameter to the set of parameters commonly used in the scheduling algorithms that are implemented by the private cloud providers. Performance tests showed that the BLJF algorithm succeeded in giving the QoS required for each user by distinguishing between them on the basis of levels of users.

**Keywords—** quantum time, response time, system performance, task weight, waiting time.

## I. INTRODUCTION

The use of advanced virtualization technology and equipment, the use of distributed systems, and the spread of services over the Internet; all this has led to the evolution of cloud computing. Cloud computing is a new model for business and services computing that is designed to give users of the Internet a variety of information services [1], [2].

The main goal of cloud computing is in changing the users' concerns from the preparation of the basic infrastructure (hardware and software) to focus on their business and make IT companies provide these services [3]. Services provided by the cloud computing are classifying into three types [4]:

- Infrastructure as a Service (IaaS): cloud provider provides all the hardware resources while the software applications should be regulated by customers.
- Platform as a Service (PaaS): cloud provider gives customers the capabilities of middleware to use the available platforms and IT solutions such as email and use of databases.
- Software as a Service (SaaS): cloud provider provides a wide range of low-cost applications as a form of

service to customers. Therefore, customers do not need to buy, installation and maintenance of these applications.

There is a wide range of services that are provided to users through the cloud system. The cloud systems can be classified on the basis of the provider which provides services [5], [6]:

- **Private clouds:** This type of cloud provides a high level of security. Private cloud owned by businessmen or companies for internal use. Main functions of this cloud are to store and process large data or to provide the necessary resources for the team of employees or customers of the company. Examples of private cloud: (OpenStack, VMware).
- **Public clouds:** Public clouds: public cloud is available for use by individuals or organizations on the basis of specific needs. Consequently, they are less secure than private clouds. Public cloud gives the highest level of efficiency in resource sharing among users. Examples of public cloud (Amazon Web Services, Google engine account, and Microsoft).
- **A hybrid cloud:** A combination of public and private clouds leads to create a hybrid cloud. It gives the capability to manage the internal resources of companies and some external resources.
- **Community cloud:** Cloud community located between public and private clouds, where it is targeting a specific group of consumers. Infrastructure and computing resources are exclusive of two or more organizations that have a common privacy, security, and regulatory considerations. And sharing of cloud infrastructure between these organizations is in order to support a particular community has the same concerns (for example, security and politics).

Cloud services that provide to Internet users are growing rapidly to give them a wide ability to do a variety of tasks. The availability of high-speed Internet and the huge number of virtual resources, which are provided by the Cloud Servers, led to increasing the number of users who use these services. Therefore, it became necessary to find a good scheduling algorithm for managing users' tasks, which are submitted to the cloud servers, in a manner to ensure the best quality of service provided to each user and maintain the good performance of the server cloud [7].

Private cloud used to protect companies, under the control of the IT department and a high level of security. Private cloud is an infrastructure that is based on dedicated hardware under control by the institution to provide services on-demand through the self-service portal. The pool of

Manuscript received February 22, 2017.

Dr. Mohammed A. Fadhil Al-Husainy is with the Department of Computer Science, Faculty of Information Technology, Middle East University, Amman, Jordan. E-mails: dralhusainy@gmail.com, malhusainy@meu.edu.jo, dralhusainy@yahoo.com.

resources provided by the private cloud inaccessible except by a single institution with greater privacy. This is the difference between private cloud and other cloud models [8], [9].

Tasks scheduling is to determine the order of the users' tasks to allocate the required resources for executing these tasks [10]. Several scheduling algorithms developed to manage users' tasks that are submitted to the cloud system such as First-Come-First-Serve (FCFS), Round-Robin (RR), Shortest-Job-First (SJF), Min-Min algorithm, and Max-Min algorithm [11]. The main goals of any scheduling algorithm in cloud computing are [5], [12]:

- Reduce the average response time by the cloud system to the submitted tasks,
- Reduce the average waiting time spent by the submitted tasks in the queue,
- Improve the use of resources in the cloud system,
- Provide highest QoS for each user in the cloud system.

All scheduling algorithms focus on giving a kind of priority for each user task, based on some factors, to arrange them in a queue for execution. This comes from the fact that: some of the tasks need to be executed first before other tasks that can wait for a longer time. This must be done to ensure the highest level of quality service that each user receives without affecting the overall system performance. A private cloud system is similar to a company system where employees who work in it are classified at different levels. Usually, the company has a top manager, a group of sectors managers and each sector manager oversees a group of employees. Therefore, private cloud environments need to use different scenarios to schedule users' tasks to achieve specific requirements for users in each level [13]-[16]. From this need, the idea emanated in this work to develop scheduling algorithm, in a private cloud, takes into account the levels of users in the institution, in addition to the other commonly used parameters in scheduling tasks, when arranging tasks in the queue for execution.

## II. RELATED WORKS

Different algorithms have been developed by researchers to schedule tasks in the cloud environment. These algorithms give high priority to one class of tasks on others based on certain parameters. All these algorithms are designed to achieve the highest QoS to users and keep the performance of the cloud system at a high level.

L. Guang, Chen-Yang, Daoguoli, [17] proposed an algorithm to schedule tasks on the basis of multi-QoS constraints and the use of the genetic algorithm. This algorithm gives users the ability to choose different scheduling goals according to their own needs. The algorithm arranges users' tasks based on the comprehensive index value  $M(task_i)$ , which represents the sum of the weights of completion time, cost, and load.

$$M(task_i) = \omega_1 \times Time + \omega_2 \times Cost + \omega_3 \times Load \quad (1)$$

Where:  $\omega_1 + \omega_2 + \omega_3 = 1$ , Time refers to the task completion time, Cost refers to the price of resources used, and Load refers to the system load.

V. L. Atul, K. Y. Dharmendra [18] suggested multi-

objective tasks scheduling algorithm, in cloud SaaS environment, to assign tasks to Virtual Machines (VMs) in order to reduce the cost and improve the throughput of the datacenter without negatively affecting the SLA (Service Level Agreement) of the application. The algorithm depends on a number of various criteria such as execution time, cost, the bandwidth of the user, etc. that are used in the cloud environment. The recorded results showed good performance and throughput of the system.

S. Selvarani, G. Sudha [19] launched from the fact that the cost of each task in the cloud environment is different from other tasks. Therefore, they proposed an algorithm depends on creating groups of tasks based on the costs of using particular cloud resources. The algorithm enhanced the communication between tasks within each group and helped to provide the required resources for all tasks within each group at the same time.

A. Singh, S. Sahu, K. Gautam, M. Tiwari [20] assumed that there are basically two groups of users in the private cloud: The first group is the most important, which includes the control and some other important functions in the institution, and the second group includes daily routine tasks. The researchers tested the throughput of the private cloud using SJF algorithms and the bounded waiting technique to solve the problem of starvation. Moreover, to maintain load balancing in a cloud system, they monitored periodically the load of the system and sent a task to the least-loaded VM.

X. Baomin, Z. Chunyan, H. Enzhao, H. Bin [21] in their paper proposed an algorithm for scheduling tasks based on Berger model. This model provides dual fairness constraints. One for the classification of users' tasks depending on QoS constraint with the exception to limit the equity of the resources in the selection of tasks. In addition, another constraint is used to judge the fairness of the resources allocation.

K. Geetinder, K. Sarabjit [22] took into account both the computation time and the transmission cost to improve the performance of Hyper-Heuristic Scheduling Approach in scheduling cloudlet and resources. The recorded results of the experiments showed that the enhanced approach succeeded to achieve better performance in terms of reducing time-period.

As noted in the works of the researchers above, the researchers chose one or more of the parameters that affect the cloud environment, such as Completion Time, Cost, and Load to achieve these highest QoS to users and keep the system performance at an acceptable level. But these researchers have ignored, in their algorithms, the fact that there are different levels of users working in the institution and they use the same private cloud environment [23]. The proposed new scenario in this paper focuses on the need to develop an algorithm to schedule users' tasks, in the private cloud environment, which takes into account the levels of users in the institution, in addition to the other commonly used parameters in scheduling tasks, by adopting a new parameter (Level-of-User) in arranging tasks in the queue for execution.

### III. THE PROPOSED BLJF ALGORITHM

Different levels/types of users in the same institution are sharing resources in a private cloud. Usually, the private cloud system used different scenarios to schedule tasks in a way that meets users' requests. In the private cloud, high-level users should enjoy special services differ from users at a low-level. Therefore, the private cloud system needs to implement effective scheduling algorithm that has the ability to give the desired QoS for each level of users.

New parameter "Level-Of-User" is used in the proposed algorithm BLJF to classify tasks of users submitted to the private cloud system. This parameter helps to distinguish between different levels of users in the institution and then give each user the appropriate QoS to execute his/her tasks.

In BLJF algorithm, five parameters are taken into account:

- **Level-of-user:** refers to the rank of the user (users' tasks) in the institution. This parameter plays the main role in the algorithm BLJF to achieve the appropriate QoS for each user who executes the tasks in the private cloud system.
- **Arrival Time:** refers to the submission time of the task to the cloud system.
- **Completion Time:** refers to the time remaining to complete the task.
- **Cost:** refers to the price of the use of resources by the task.
- **Load:** refers to the load of the private cloud system.

The algorithm BLJF starts from the assumption that the private cloud system provides service to a number of users of different levels: (1...N), where level 1 represents the user at the highest level and level N represents the user at the lowest level.

Weight is assigned to each task submitted to the private cloud server. This weight is used by the algorithm BLJF to arrange all tasks in the queue. The weight of the task is a list of five elements (numbers/values)  $V_1, V_2, V_3, V_4,$  and  $V_5$  arranged in the list in a certain sequence depending on the factor  $S_x$  of each value  $V_x$  ( $x$  within 1...5). The value of each  $S_x$  is determined by the institution. Where:

$$V_1 = (\text{Level-Of-User})_{s_1}$$

$$V_2 = (\text{ArrivalTime})_{s_2}$$

$$V_3 = (\text{CompletionTime})_{s_3}$$

$$V_4 = (\text{Cost})_{s_4}$$

$$V_5 = (\text{Load})_{s_5}$$

And  $S_x$  is an integer number within one of two cases:

- Case 1:** If  $S_x = 0$ , means that the  $V_x$  (i.e., the parameter related) is not used in calculating the task weight.
- Case 2:** If  $S_x \neq 0$ , means that the  $V_x$  (i.e., the parameter related) used in calculating the task weight. Each  $S_x$  between (1...5) and has a different value than others. Where:
- If  $S_x = 1$ , means that the  $V_x$  (i.e., the parameter related) is arranged as the *first* element in the weight's list.
  - If  $S_x = 2$ , means that the  $V_x$  (i.e., the parameter related) is arranged as the *second*

element in the weight's list.

- If  $S_x = 3$ , means that the  $V_x$  (i.e., the parameter related) is arranged as the *third* element in the weight's list.
- If  $S_x = 4$ , means that the  $V_x$  (i.e., the parameter related) is arranged as the *fourth* element in the weight's list.
- If  $S_x = 5$ , means that the  $V_x$  (i.e., the parameter related) is arranged as the *fifth* element in the weight's list.

Examples:

1. If  $S_1=1, S_2=4, S_3=3, S_4=5,$  and  $S_5=2$ . This means that the sequence  $V_1, V_2, V_3, V_4,$  and  $V_5$  in the list of task weight is:

$$\text{Task Weight: } \begin{array}{|c|c|c|c|c|} \hline V_1 & V_5 & V_3 & V_2 & V_4 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

2. If  $S_1=1, S_2=3, S_3=0, S_4=2,$  and  $S_5=0$ . This means that the sequence  $V_1, V_2, V_3, V_4,$  and  $V_5$  in the list of task weight is:

$$\text{Task Weight: } \begin{array}{|c|c|c|c|c|} \hline V_1 & V_4 & V_2 & & \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

The tasks are arranged in the queue according to the sequence  $V_x$  in the weight's list. Where, the tasks are arranged firstly on the basis of  $V_x$  in the element 1, and secondly on the basis of  $V_x$  in the element 2, and so on.

The proposed algorithm BLJF keeps  $V_1$  in the element 1 of the list of task weight in order to make the order of tasks in the queue depends mainly on the 'Level-Of-User' parameter because this is the main goal of the algorithm BLJF. This will lead to arrange the tasks of users in the queue based on the levels of users in the institution.

In the algorithm BLJF, all tasks are arranged in a circular queue. Small units of equal time called 'time slice' or 'quantum' are given to each task periodically to be executed until complete the task. The quantum is calculated by finding the Median of the values of time remaining to complete each task in the queue.

The following are the main steps of the proposed algorithm BLJF that are implemented in the private cloud system:

- Step 1:** Determine the factors  $S_1, S_2, S_3, S_4,$  and  $S_5$  for the values  $V_1, V_2, V_3, V_4,$  and  $V_5$  through cooperation between the private cloud system and the institution.
- Step 2:** Create the list of task's weight for each task in the queue by arranging  $V_1, V_2, V_3, V_4,$  and  $V_5$  in the list.
- Step 3:** Determine the value of quantum by finding the Median of the values of time remaining to complete each task in the queue.
- Step 4:** Arrange all the tasks in the queue (in ascending order) based on the lists of the tasks weights in Step 2.
- Step 5:** Give period of time (equivalent to quantum) to the next task in the queue to be executed.
- Step 6:** If there are new tasks submitted by the users, go to Step 2. Otherwise, go to Step 5.

IV. PERFORMANCE EVALUATION OF THE ALGORITHM BLJF

CloudSim software has been used to generate a large set of random tasks (data set) of the different values of the five parameters. These tasks have been used in the experiments to evaluate the performance of the algorithm BLJF. The data set used in the experiments consist of more than 500 tasks are randomly generated and submitted to the private cloud system at different times. This is done on the assumption that there are 10 levels of users in the institution who use the private cloud system and  $S1=1$ ,  $S2=3$ ,  $S3=2$ ,  $S4=4$ , and  $S5=5$ .

Experiments, in this work, been implemented using the computer system has the following specification:

- Processor: Intel core i3-3110M CPU, 2.40GHz
- Installed memory (RAM): 4.00 GB
- Hard disk drive: 500 GB
- Operating System: 64-bit Windows 7

Two measurements commonly used to evaluate the performance of any tasks scheduling algorithm are the Average Response Time (ART) and the Average Waiting Time (AWT) for the submitted tasks. Where, any good scheduling algorithm is the one that achieves lowest values for both ART and AWT which means providing the highest QoS to users. These two measurements have already been used in this work to evaluate the performance of the proposed algorithm BLJF. Use these measurements, in the comparison between any proposed algorithm with other known standard scheduling algorithms (FCFS, SJF and RR), helps to evaluate the performance of the proposed algorithm. Therefore, the performance of the most tasks scheduling algorithms developed by researchers is evaluated by conducting this comparison.

Fig. 1 shows the recorded values of AWT and ART (in millisecond) for the algorithm BLJF (depending on the levels of users in the institution) and the standard algorithms (FCFS, SJF and RR).

It is clear from Fig. 1 that the algorithm BLJF succeeded to prioritize between the different levels of users in the institution who use the same private cloud system. This has been done by:

- Reducing the waiting time (in the queue) for users at higher levels at the expense of users at lower levels.
- Speed up the response time (by the private cloud system) to users at higher levels at the expense of users at lower levels.

This leads to give each user the required QoS based on the level of the user in the institution. But this must not affect the overall performance of the cloud system.

Therefore, to give a closely look at the overall performance of the algorithms: BLJF, FCFS, SJF and RR, Fig. 2 shows the total AWT and total ART for the four algorithms. In Fig. 2, we can note that the use of the new parameter 'Level-Of-User' in scheduling tasks of users helped in giving the algorithm BLJF the ability to:

- Maintain the total AWT at an acceptable level, compared with other standard algorithms (FCFS, SJF and RR).
- Achieve a minimum total ART compared with other

standard algorithms (FCFS, SJF and RR).

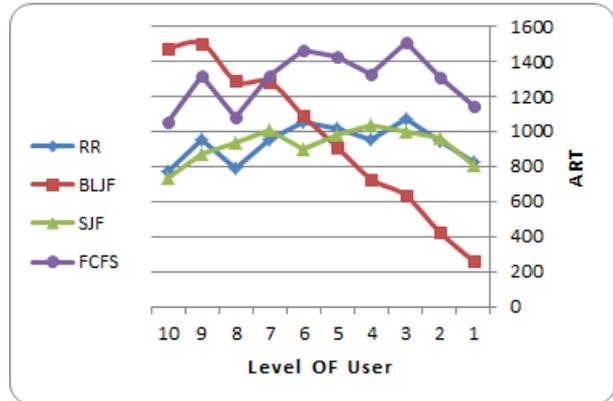
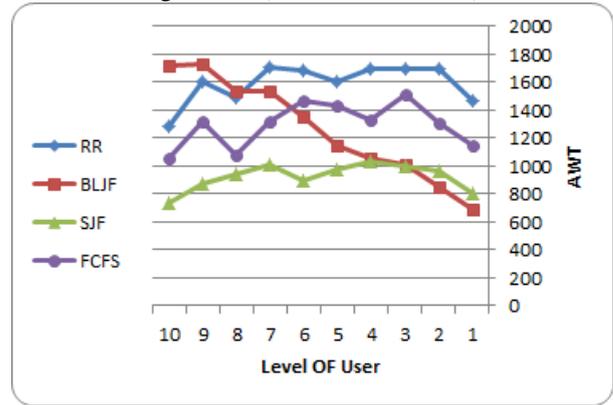


Figure 1. AWT and ART for BLJF, FCFS, SJF and RR algorithms.

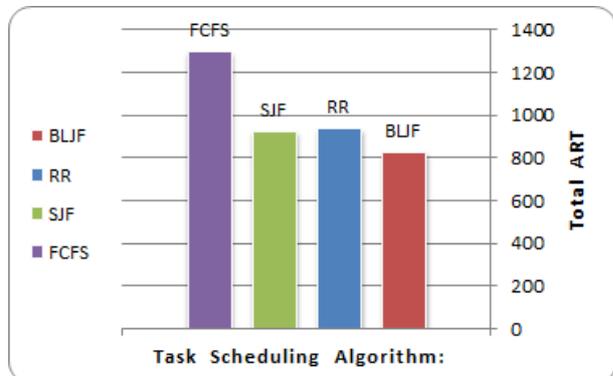
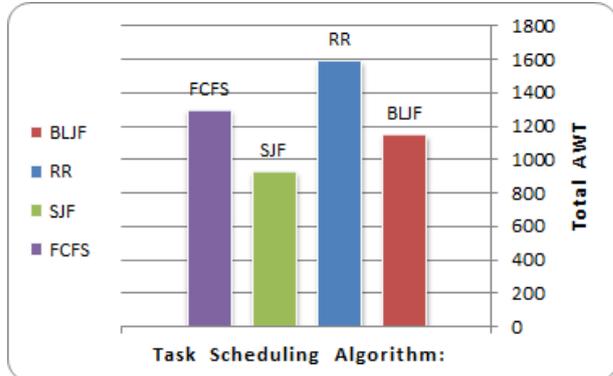


Figure 2. Total AWT and ART for BLJF, FCFS, SJF and RR algorithms.

To highlight the performance of the algorithm BLJF when changing the number of levels of users in the institution, Fig.

3 shows the total AWT and total ART for a different number of levels of users.

It is clear from Fig. 3 that:

- When there is a few number of levels of users in the institution, this means that there are significant differences between the levels of users in the institution (i.e., top manager and employees). Therefore, it becomes necessary to distinguish between the QoSs that are given to the users in each level. This leads to increase the disparity and instability in the results recorded.
- When the number of levels of users increased in the institution (which it is normal), this means that there are small differences between the levels of users in the institution. Therefore, the QoSs that are given to the users in each level become close to each other relatively. This leads to decrease the disparity and greater stability in the results recorded.

On the other hand, to stand on the performance of the algorithm BLJF when changing the number of tasks submitted, Fig. 4 shows the total AWT and total ART (that have been recorded for the four scheduling algorithms BLJF, RR, SJF, and FCFS) for a different number of tasks using same number of levels of users (10 levels of users).

As in other standard scheduling algorithms known (RR, SJF, and FCFS), changes in the total AWT and total ART of the algorithm BLJF increase proportionally with the increase in the number of tasks submitted to the cloud server. This means that the adoption of the 'Level-Of-User' in the algorithm BLJF does not affect negatively on the algorithm and does not make it behave differently from standard scheduling algorithms.

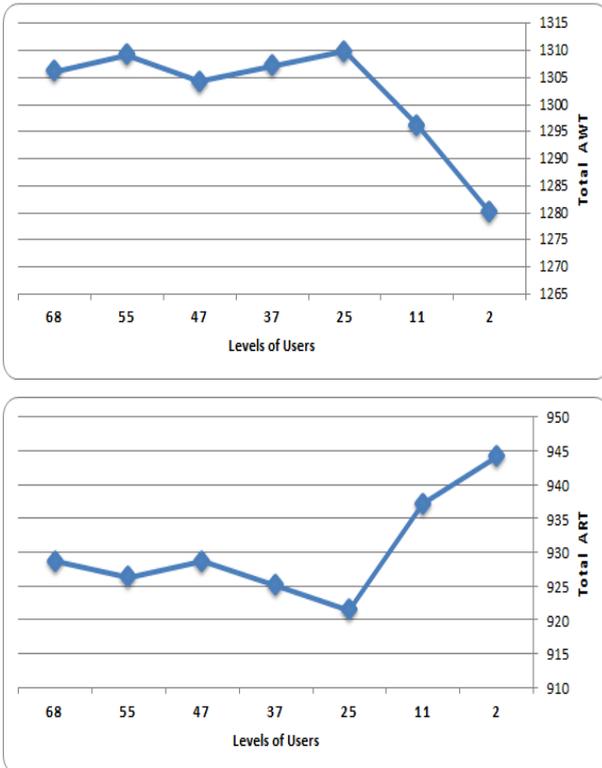


Figure 3. Total AWT and ART of BLJF algorithm for the different number of levels of users.

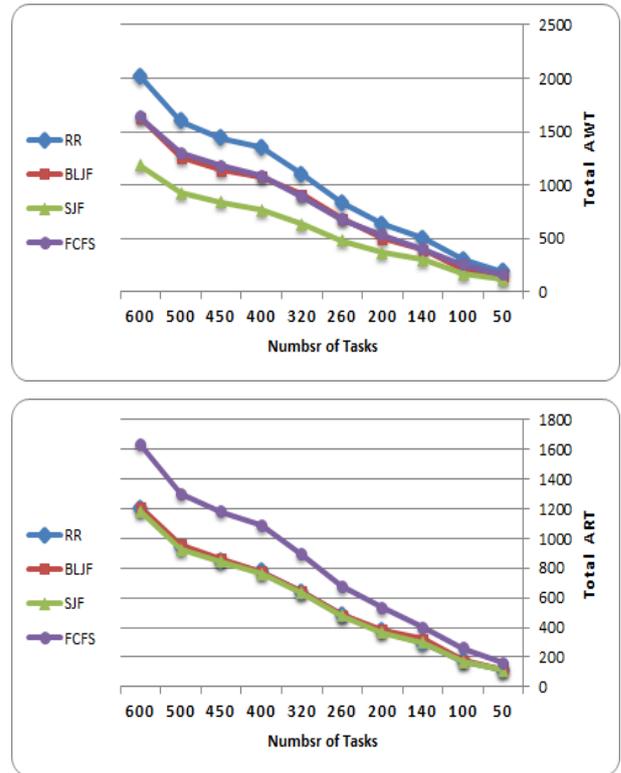


Figure 4. Total AWT and ART of BLJF, RR, SJF, and FCFS algorithms for different number of tasks

In addition, the algorithm BLJF has the ability to change the behavior of its work easily by:

- Assign a zero value or a non-zero value in one or more of S1, S2, S3, S4, and S5 to calculate the  $V_1, V_2, V_3, V_4,$  and  $V_5$ .
- Change the sequence of  $V_1, V_2, V_3, V_4,$  and  $V_5$  in the list of the task weight by changing the non-zero values of the S1, S2, S3, S4, and S5.

The algorithm can dominate or neutralize the effect of any parameter in the calculation of the task weight depending on certain conditions and policies that must be followed in the private cloud system. This will change the values of the tasks' weights and certainly lead to change in the way used by BLJF to arrange these tasks in the queue for execution. Set of experiments has been implemented using different values of the S1, S2, S3, S4, and S5 to test the ability of the algorithm BLJF. Table I and Fig. 5 show the values of the S1, S2, S3, S4, and S5 used in the experiments and the total AWT and total ART values recorded in these experiments. It is clear from Fig. 5 that the algorithm BLJF can change the behavior of its work when it depends on one of the five parameters and ignores the others, this will produce different results and lead to give different QoS for users of private cloud system in each case.

Table I: Set of Experiments Uses One of the Five Parameters and Cancels others Parameters.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
<b>S1:</b> Levels Of Users	1	0	0	0	0
<b>S2:</b> Arrival Time	0	1	0	0	0
<b>S3:</b> Completion Time	0	0	1	0	0
<b>S4:</b> Cost	0	0	0	1	0
<b>S5:</b> Load	0	0	0	0	1
<b>Total AWT</b>	1650.45	1598.91	923.91	1609.12	1668.04
<b>Total ART</b>	903.24	936.92	923.91	914.65	910.85

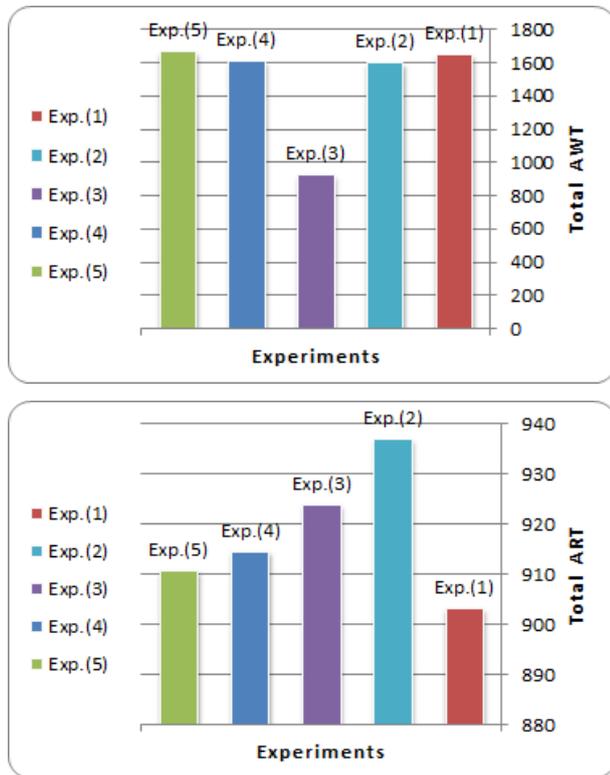


Figure 5. Total AWT and ART of BLJF algorithm of the experiments in Table I.

## V. CONCLUSION

'Level-Of-User' has been used by the algorithm BLJF as a new parameter in addition to the four parameters commonly used in the cloud system. The adoption of 'Level-Of-User' as the main parameter, in arranging tasks of users in the queue to be executed, led to redistribute the services, provided by the private cloud system, to ensure giving each user in the institution the QoS that is deserved. Furthermore, the representation of the task weight in the form of an ordered list of the parameters  $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$ , and  $V_5$  helped the BLJF:

- To control the effect of each parameter on the scheduling of tasks in the queue
- Allowing the use or ignore any one of the five parameters.

The comparison tests of the algorithm BLJF with the other standard scheduling algorithms showed that the algorithm BLJF can be used effectively for scheduling tasks in the private cloud and achieving highest QoS provided to the users based on their levels in the institution.

## REFERENCES

- [1] A. Ioannis, D. Helen, "Evaluation of gang scheduling performance and cost in a cloud computing system," *The Journal of Supercomputing*, vol. 59, no. 2, pp.975-992, 2010. doi:10.1007/s11227-010-0481-4
- [2] L. M. Mustafa, M. K. Elmahdy, M. H. Haggag, "Improve scheduling task based task Grouping in cloud computing system," *International Journal of Computer Applications*, vol. 93, no. 8, pp. 0975 – 8887, 2014. doi: 10.5120/16232-5561
- [3] R. Buyya, R. Ranjan, R. N. Calheiros, "Modeling and Simulation Of Scalable Cloud Computing Environments And The cloudsim toolkit: challenges and opportunities," *High Performance Computing & Simulation HPCS'09*, pp. 1-11, 2009. doi: 10.1109/HPCSIM.2009.5192685
- [4] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41 no. 1, pp. 23-50, 2011. doi:10.1002/spe.995
- [5] S. Goyal, "Public vs private vs hybrid vs community - cloud computing: a critical review," *I.J. Computer Network and Information Security*, vol. 3, pp. 20-29, 2014. doi: 10.5815/ijcnis.2014.03.03
- [6] M. Katyal, A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment," *International Journal of Distributed and Cloud Computing*, vol. 1, no. 2, 2013. (doi not available)
- [7] B. Rajkumar, S. Y. Chee, V. Srikumar, B. James, B. Ivona, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, pp. 599-616, 2009. doi=10.1.1.144.8397
- [8] K. Subramanian, Private clouds, Trend Micro Inc, 2011.
- [9] Navimipour, N. J., & Milani, F. S. "Task scheduling in the cloud computing based on the cuckoo search algorithm," *International Journal of Modeling and Optimization*, vol. 5, no. 1, 44, 2015. doi: 10.7763/IJMO.2015.V5.434
- [10] S. Lovesum, K. Krishnamoorthy, P. Prince, "An optimized QoS based cost effective resource scheduling in cloud," *Journal of Theoretical & Applied Information Technology*, vol. 66, no. 1, 2014. (doi not available)
- [11] E. Shimpy, Mr. J Sidhu. "Different scheduling algorithms in different cloud environment," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3 no. 9, pp. 8003-8006, 2014. (doi not available)
- [12] M. A. F. Al-Husainy, "Best-Job-First CPU scheduling algorithm," *Information Technology Journal*, vol. 6, no. 2, pp. 288-293, 2007. doi: 10.3923/itj.2007.288.293
- [13] L. Yang, C. S. Panb, E. H. Zhanga, H.Y. Liua. "A new class of priority-based weighted fair scheduling algorithm," *Physics Procedia*, vol. 33, pp. 942–948, 2012. doi: 10.1016/j.phpro.2012.05.158
- [14] P. Shachee, S. Richa, "Double level priority based optimization algorithm for task scheduling in cloud computing," *International Journal of Computer Applications*, vol. 62, no. 20, pp. 33-37, 2013. doi: 10.5120/10213-5051
- [15] G. Shamsollah, M. Othman, "A priority based job scheduling algorithm in cloud computing," *International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012)*. Procedia Engineering vol. 50. pp. 778–785, 2012. doi: 10.1016/j.proeng.2012.10.086
- [16] P. Swachil, B. Upendra, "Priority based job scheduling techniques in cloud computing: a systematic review," *International Journal of Scientific & Technology Research*, vol. 2, no. 11, 2013. (doi not available)
- [17] L. Guang, Chen-Yang, Daoguoli, "Scheduling research based on genetic algorithm and QoS constraints of cloud computing resources," *Journal of Theoretical & Applied Information Technology*, vol. 51, no. 1, 2013. (doi not available)
- [18] V. L. Atul, K. Y. Dharmendra, "Multi-objective tasks scheduling algorithm for cloud computing throughput optimization," *International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015)*. Procedia Computer Science vol. 48, pp. 107–113, 2015. doi: 10.1016/j.procs.2015.04.158
- [19] S. Selvarani, G. Sudha, "Improved cost-based algorithm for task scheduling in cloud computing," *Proc. of the IEEE Int. Conf. on Computational Intelligence and Computing Research, ICCIC*, pp. 1-5, 2010. doi: 10.1109/ICCIC.2010.5705847

- [20] A. Singh, S. Sahu, K. Gautam, M. Tiwari, "Private cloud scheduling with SJF, bound waiting, priority and load balancing," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 1, pp. 367-372, 2014. (doi not available)
- [21] X. Baomin, Z. Chunyan, H. Enzhao, H. Bin, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software* vol. 42, pp. 419-425, 2011. doi: 10.1016/j.advengsoft.2011.03.007
- [22] K. Geetinder, K. Sarabjit, "Improved hyper-heuristic scheduling with load-balancing and RASA for cloud computing systems," *International Journal of Grid and Distributed Computing*, vol. 9, no. 1, pp.13-24, 2016. doi: 10.14257/ijgcd.2016.9.1.02
- [23] I. A. Mohammed, "Task scheduling using best-level-job-first on private cloud computing," *M.Sc. thesis*, Middle East University, 2016. (doi not available)

#### Author



**Mohammed Abbas Fadhil Al-Husainy** received the M.Sc. and Ph.D. degrees in 1996 and 2002, respectively. From 1997 to 2002, he was a lecturer in Department of Computer Science, Al-Hadba University of Mosul. From 2002 to 2013 he has been an associate professor in Departments: Computer Science and Multimedia

Systems, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan. Since 2014, he has been an associate professor in Department of Computer Science, Faculty of Information Technology, Middle East University. He taught many courses such as microprocessors, data structures, algorithm design and analysis, digital design systems, operating systems, cryptography, computer organization, programming languages. His research interests are in the broad field of algorithm design, including multimedia data processing, scheduling algorithms, and cryptography algorithms.