ISSN 2278-3091
Volume 8, No.1.4, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse2781.42019.pdf
https://doi.org/10.30534/ijatcse/2019/2781.42019

# OTS: An Optimal Tasks Scheduling Algorithm Based on QoS in Cloud Computing Network

**Mohammed Ameen Alhakimi[1, 2], Rohaya Latip[2, 3]**

[1]Ministry of Higher Education and Scientific Research, Yemen
[2]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia.
[3]Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia.
mohammedalhakimi@hotmail.com
rohayalt@upm.edu.my

## ABSTRACT

Cloud Computing has emerged as a service model that offers online accessible resources to the clients. These resources contain storage, servers, and other applications and it provides security, flexibility, and scalability. In Max-Min algorithm where the large tasks have their priority to be scheduled first this leads small tasks to stay longer in the queue until all huge tasks finished their execution. This study presents an optimal tasks scheduling algorithm by enhancing Max-Min algorithm. The simulation results have proven that the Proposed Optimal Tasks Scheduling OTS completes tasks execution with less execution time and higher performance compared with Max-Min and TS algorithms. The overall results show that the performance of the proposed algorithm achieved 6% better in terms of time execution compared of both of Max-Min and TS algorithms

**Key words:** Cloud computing; scheduling; OTS; QoS; Max-Min.

## 1. INTRODUCTION

The concept of Cloud Computing leads many business and organizations to turn toward using this technique. There is nothing essentially new in any of the technologies that make up cloud computing as most of these technologies have been used before.Numerous experts in the academic field have attempted to define precisely what “cloud computing” is and what unique attributes that cloud computing presents. Cloud computing is an addition of different techniques such as parallel, distributed, and grid computing, it is a recent model for enabling access to a shared computing, storage resources, and services which can be accessed through the internet using a set of applications. Virtualization, dispersal and dynamic extensibility are the basic features of this environment [1].

“Software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS)” are three dissimilar services where it can be provided by cloud computing. Clients can submit their works into the cloud for computational transmutation or rather leave the data in the cloud for future use. Therefore, different users may have different requirements. One of the major requirement in cloud computing is to assign the job to appropriate resources while taking the QoS into consideration. [2].

To satisfy the requirement of this dynamic environment, a lot of the researchers been worked in this area and many algorithms proposed which meet this requirement. In cloud computing where service and storage can be access by the client, Clients can submit their tasks into the c1oud for computation processing purpose or rather for future storage [7]. Those clients who are using this service have different requirements, this requirement knows as Quality of Service, which is a cooperative effort of service performance that defines the degree of gratification of a user for the service [2]. The dynamic nature of this environment can endure a big challenge as some assigned resources might overload during the execution that may cause the failure of the whole execution process [8]. Hence, Scheduler in the cloud must have the ability to schedule clients’ tasks/jobs in such a way that cloud provider can earn a maximum advantage for his service [7]. In Max-Min algorithm where the large tasks have their priority to schedule first, this leads small tasks to stay longer in the queue until all large length tasks finished their execution. Thus, mapping this task to the resource should take it into the consideration in order to increase the performance of this environment.

An optimal task scheduling algorithm OTS is one of the algorithms which execute the tasks based on its attribute then schedule the tasks with the resource that will take less execution time.The focus of this research is to implement an algorithm for tasks scheduling in cloud computing environment which can solve the problem of time execution for all tasks to satisfy users need. The rest of the paper is structured as follows: Section 2 describes the related work, Section 3 explains the proposed algorithm, Section 4 declares the experimental setup, in section 5 we explain the simulation analysis and the performance of the proposed algorithm, Section 6 concludes this paper it identifies future work and possible next step in research.

## 2. RELATED WORKS

Cloud computing defined as a new business model for providing services and resources through the internet. It is a combination of several concepts from virtualization, distributed application design, grid computing, utility computing, and clustering [3]. In this paper, we have done an extensive survey on several scheduling algorithms which minimizes the total execution time. The following are the essential algorithms that considered for this study:

In Grouped tasks scheduling [2] the author proposed an algorithm which is a composed of two methods Min-Min and Task scheduling TS. The main idea here is to apply the QoS to achieve a less time of execution to all tasks as min-min algorithm has achieved with low latency to tasks with high priority in order to satisfy user's need. In this model, scheduling done in two phases. First is to decide which category should schedule first and this will be based on the attribute of the tasks, therefore the task that has high attribute will lead that category to schedule first. In this step, GTS algorithm will use the same way of TS algorithm to calculate the priority of each category. Second is to decide which task inside the chosen category will schedule first and this will be based on the execution time of task inside the chosen category.

In [4], the authors proposed a "task scheduling algorithm based on QoS-driven" called TS_QoS which compute the precedence of tasks according to four attributes (user privilege, the expected scheduled priority of tasks, length of tasks, and tasks latency) then tasks will be sorted into a service which can execute the task quickly according to the sorted task queue. [3] Min-Min algorithm works on the execution time of tasks, therefore tasks that have less time execution will be scheduled first while tasks with maximum execution will suffer until all small length tasks finished their execution [5]. The scheduling here done in two phases, the first phase is to find the minimum execution time of all tasks by allocating them to the resources that can complete them in the shortest possible time. The Second phase is to choose tasks with the smallest time execution amongst all tasks. This algorithm proceeds by allocating the task to the resource that produces the smallest time completion.

Max-Min algorithm depends on the execution time, therefore tasks that have maximum execution time will schedule first while tasks with minimum length will wait in the queue until all huge length tasks finish their execution. Similar to Min-Min algorithm, not all tasks are scheduled at the beginning. A resource that takes minimum time for task completion is chosen. Then the task with large completion time is mapped to the selected resource. All the large tasks are executed first. Therefore, large tasks do not have to wait for a long time for execution [6].

## 3. PROPOSED OPTIMAL TASKS SCHEDULING

The idea of the proposed model works in two phases. In the first, classify the task by computing its priority based on its attribute. The attributes of tasks are utilized as average and long length tasks. Secondly, the resources *Rj* is isolated into two groups slow and fast resources. This apportioning is done by arranging the resources in ascending order from the slowest to the fastest resources. When tasks arrived for execution and to reduce the completion time of it, the resource that takes minimum execution time will be selected *Ri* among all resources *Rj*. if the selected resource *Ri* from the first group, then map the task *Ti* with average length attribute to it else the task *Ti* with long length attribute will be mapped to the selected resource.

In our method, we used the model of packet simulation, which made up of multiple transmitting nodes, single queue, and multiple destinations. The proposed model to use in this simulation as shown in Figure 1.

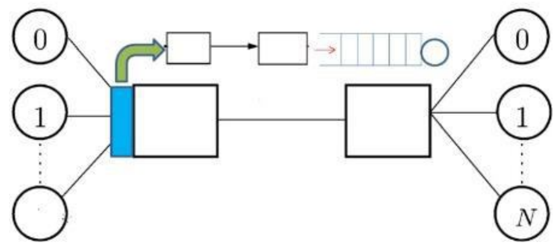


**Figure 1**: Model of packet simulation

**Packet generation**: is an event *e1* when the packet or task arrive to the queue in order to be scheduled.

**Packet departure**: is an event *e2* when the packet transmitted exit the queue and arrive the available server.

## 4. EXPERIMENTAL SETUP

To evaluate the proposed algorithm, we developed an extensive simulation platform based on CloudSim. To simulate cloud environment we have considered a Data Center and Scheduler. The execution time for all of Max-Min, TS and the proposed tasks scheduling algorithms have tasted with 200-1000 tasks in 50 or 100 VMs respectively.

**Table 1:** Simulation parameters [3], [2]

| Simulation Parameters | Value |
|---|---|
| Number of tasks | 200, 400, 800, 1000 |
| Number of VMs | 50, 100 |
| No. running times | 10 times |
| Data center | 1 |
| Scheduler | 1 |

In our simulation experiment, we test the execution time span for three algorithms with 200-1000 tasks in 50 or 100 Visual machines respectively. The reason behind choosing these workflows is that previous research papers [4], [2] have used these datasets to test both of Tasks scheduling TS and Grouped tasks scheduling algorithms GTS. Therefore, one data center, one scheduler have been used for this experiment.

The purpose of this experiment is to demonstrate that the performance of the proposed algorithm is better than Max-Min and Tasks scheduling algorithms. For the simulation purpose, Max-Min algorithms have been developed in Java using their pseudo codes. The reasons for developing those existing algorithms is that because we compare our proposed algorithm with both of Max-Min and Tasks Scheduling to check whose performance is better.

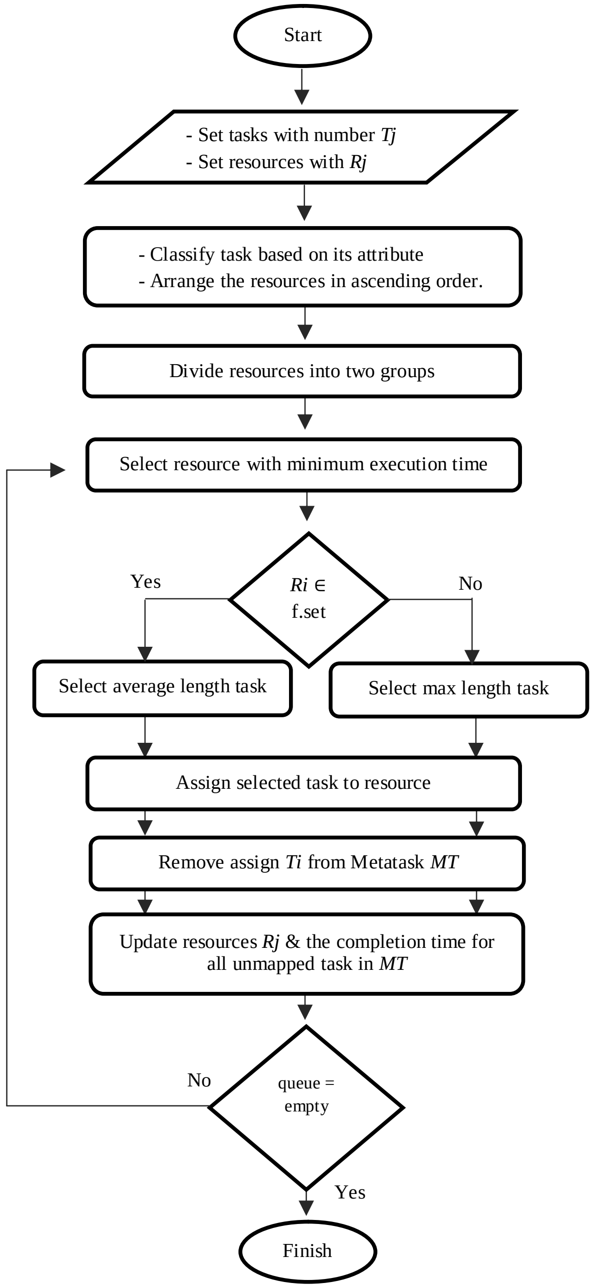


**Figure 2**: Flowchart of the proposed algorithm

In this experiment, we tested the three algorithms using time span where it defined as the time taken from the processing start until the last task finish its process.

$$Execution\ time = Te - Ts \quad (1)$$

where: Te = the finish time for the last task
Ts = the time of start first task.

## 5. RESULT ANALYSIS

In this section, we want to verify the effectiveness of the proposed algorithm. We have conducted several experiments to test the performance of the proposed algorithm in terms of execution time and this section shows performance comparison between the proposed algorithm with the existing scheduling algorithms Max-Min and TS algorithms.

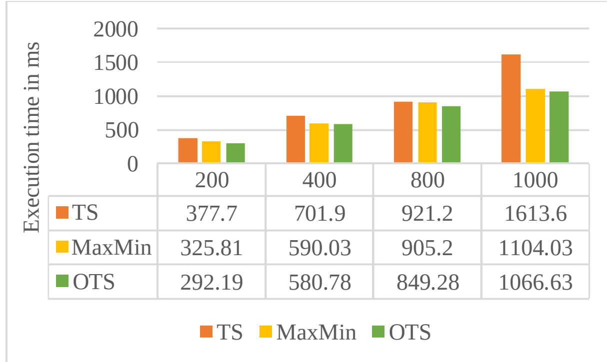


**Figure 3**: Execution time in ms with number of services 50 VMs.

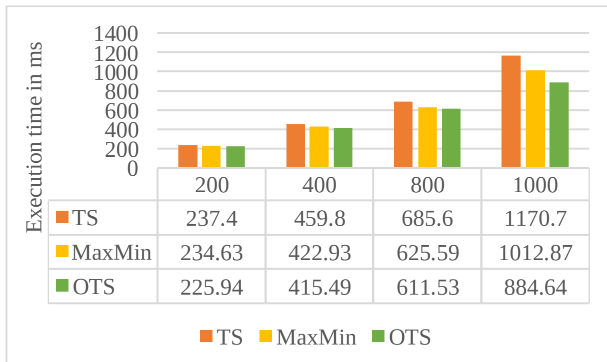


**Figure 4**: Execution time in ms with number of services 100 VMs.

Results of above evaluations show that the proposed OTS completes tasks execution with less execution time and higher performance compared with both Max-Min and TS algorithms. Performance of proposed algorithm is around 5% percent better than Max-Min algorithm and around 19% percent better than TS algorithm for 50 virtual machines. Performance of proposed algorithm is around 5% percent better than Max-Min algorithm and around 10% percent better than TS algorithm for 100 virtual machines.

Results show that the proposed algorithm behaves better in terms of time execution after testing it in 10 running time with 200-1000 workflows. As we increased the number of virtual machines which are used for the simulation the proposed algorithm still acting better.

## 6. CONCLUSION

This paper presents an optimal tasks scheduling algorithm while taking the QoS into consideration. The proposed algorithm was implemented using CloudSim.

In Max-Min algorithm where a large task is assigned to the fastest resource, this large task might map to the slow resource. The limitation here leads to increase the total length of schedule. But in the proposed algorithm positively, schedule long length tasks to the slow resource has been reduced. The solution used in this paper is solved by dividing resources into two different groups according to MIPS speed. If the fastest available resource is from the first group which is the group of slow resources, then the average length task is mapped to it. If the fastest available resource is from the second group which is the group of fast resources, then the largest task is mapped to it.

After analyzing the simulation results we come to the conclusion that the proposed algorithm outperform both of Max-Min and TS algorithms in teams of execution time span, and well performance in terms of resource utilization. The overall results show that the performance of the proposed algorithm achieved 6% percent better in terms of time execution compared with Max-Min and TS algorithms. Therefore, further optimization can be done by: (1) Intend to increase the number of attributes that can apply in the algorithm, (2) the algorithm can work with dependent tasks and (3) the algorithm can work in real time.

## ACKNOWLEDGMENT

We thankful to Universiti Putra Malaysia for unfailing support in the successful conduction of this research work.

## REFERENCES

1. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, **Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility**, Future Generation Computer Systems,25:599-616, 2009 https://doi.org/10.1016/j.future.2008.12.001
2. Gamal El Din Hassan Ali Hend, Saroit Imane Aly, Kotb Amira Mohammed. **Grouped tasks scheduling algorithm based on QoS in cloud computing network**. In: Egyptian Informatics Journal. eij; 2016. https://doi.org/10.1016/j.eij.2016.07.002
3. Liu Gang, Li Jing, Xu Jianchao. **An Improved Min-Min Algorithm in Cloud Computing.** In: Proceedings of the 2012 international conference of modern computer science and applications, Zhenyu Du; 2013. https://doi.org/10.1007/978-3-642-33030-8_8
4. Wu Xiaonian, Deng Mengqing, Zhang Runlian, Zeng Bing, Zhou Shengyuan. **A task scheduling algorithm based on QoS-driven in cloud computing.** In: International conference on information technology and quantitative management. ITQM 2013. https://doi.org/10.1016/j.procs.2013.05.148
5. Liu Gang, Li Jing, Xu Jianchao. **An Improved Min-Min Algorithm in Cloud Computing**. In: Proceedings of the 2012 international conference of modern computer science and applications, Zhenyu Du; 2013. https://doi.org/10.1007/978-3-642-33030-8_8
6. Sindhu, S. **Task scheduling in cloud computing**, International Journal of Advanced Research in Computer Engineering Technology 4, 2015.
7. Abdullah Monir, Othman Mohamed. **Cost-based multi-QOS job scheduling using divisible load theory in cloud computing**. In: International conference on computational science. ICCS; 2013. https://doi.org/10.1016/j.procs.2013.05.258
8. Bousselmi Khadija, Brahmi Zaki, Gammoudi Mohamed Mohsen. **QoS-aware scheduling of Workflows in Cloud Computing environment.** In International Conference on Advanced Information Networking and Applications. IEEE; 2016. https://doi.org/10.1109/AINA.2016.72