Review On A Comparative Study Of Various Task Scheduling Algorithm In Cloud Computing Environment

Anuja Patil[a], Blessy Thankachan[b]

[a]Research Scholar, School of Computer and System Sciences, Jaipur National University, Jaipur, Rajasthan
[b] Associate Professor, School of Computer and System Sciences, Jaipur National University, Jaipur, Rajasthan

Abstract: Cloud computing environment is meant to be consistently offering different services at requested time. Availability and waiting time plays very important role in this. By considering these two factors, scheduling of task with efficient algorithm in cloud environment is needed to be discovered and assign that task for execution. In this environment, virtual storage, network and servers are used so resource utilization and paid services should be timely available. In this paper, we will compare different task scheduling algorithms by considering some basic factors like makespan, request time, turnaround time etc.

Keywords: Task, Scheduling, Turnaround time, Waiting time, Response time, Throughput, VM.

1. Introduction

Cloud computing is trending since 2007. As we know now everywhere digitization is must and to make things available at requested time is need of everyone. So fulfilling this cloud computing environment is design which will be provided under one roof with many facilities that is well-organized and services that are timely available. Cloud computing provides security to our data, services, applications, platform, etc. Most important part of this technology is, it should be more efficient and reliable. It's a network based technology so there are many factors like number of tasks, bandwidth, to be considered for making those services available within requested time. Operating system supports multitasking which facilitates execution of multiple tasks concurrently without waiting for another task to be completed. One of the objectives of multitasking is to maximize resource utilization, which is achieved by sharing system resources among multiple users and system processes. Efficient resource sharing depends on efficient scheduling of competing processes. Uniprocessor scheduling involves only one processor and can execute only one process at a time.

2. Significance Of The Study

The scheduling criteria includes the following:

1. CPU utilization – Engaging CPU every time is the main aim of scheduling.

2. Throughput – It is calculated by counting of work completed by CPU, number of processes execution per unit time.

3. Turnaround time – Duration between submitting the process till the process completes its execution.

4. Waiting time – The time process is staying in ready queue.

5. Response time – Duration of process submission till it generates its first response.

The following are different Task Scheduling Algorithms used in cloud environment(Sharma et al., 2019) as shown in Fig 1.

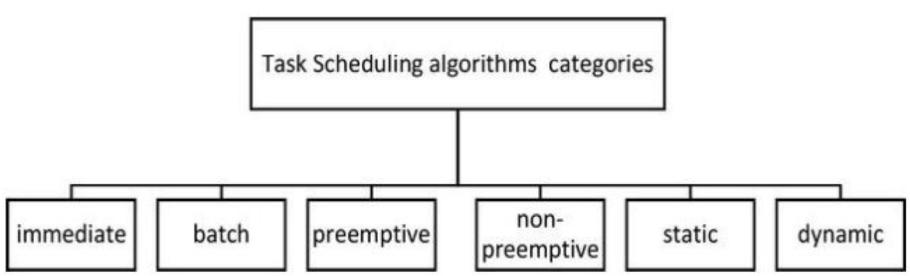

Figure.1 Categories of different task scheduling techniques in cloud environment

There are many task scheduling algorithms existing but all are not suitable for distributed environment. Considering it, American Journal of Intelligent Systems 2017, as shown in Fig 2 mainly divides the cloud-computing task scheduling into two parts.

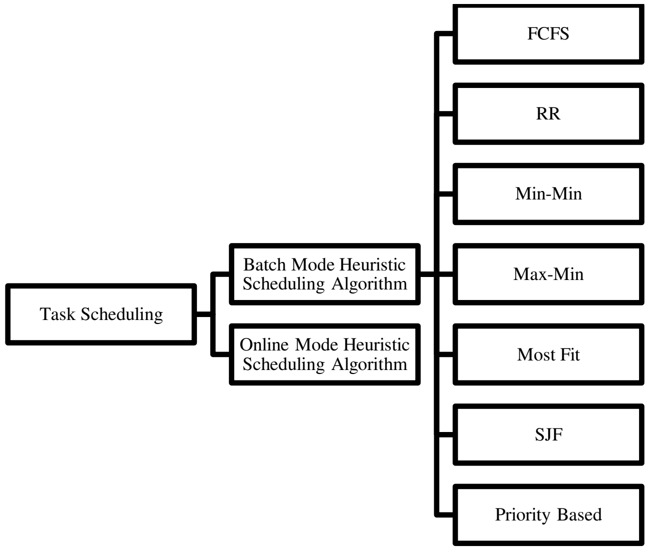

Figure.2 Different algorithms for task scheduling

a. **Batch Mode Heuristic scheduling Algorithms (BMHA):** In this type, tasks are placed in queue and collected when tasks gets into the system. Then it assigns task scheduler to execute those tasks.

i **First Come First Serve Scheduling Algorithm:** It is non-preemptive algorithm. As task enters into queue, it is assigned to the processor. It works on First In First Out manner. It is very simple and fast. However, it utilize less VM.

ii **Round Robin Scheduling Algorithm:** It is preemptive algorithm. As task enters, in that sequence only it gets chance for execution with fixed duration or time quantum but if in case that much time is not enough for executing that task then it will get preempted and placed inside ready queue to get chance for execution again. Its Response time is good, good load balancing, as compared to other algorithms it is less complex(Fataniya & Patel, 2018). However, as quantum expires process is thrown out of execution.

iii **Min-Min Scheduling Algorithm:**As task enters in queue, it compares with other task, find minimum of all, and assign that task for execution to VM. It gives better Makespan, but difficulties in load balancing.

iv **Max-Min Scheduling Algorithm:** As task enters in queue, it starts comparing with other tasks, it finds maximum out of all, and then it assigns to VM. This algorithm allow largest task to be executed first. It gives better makespan. However, task with less execution time has to wait, so it might go under starvation.

v **Most Fit Task Scheduling Algorithm:**As task enters according to execution time, it is selected according to best allocation. The task that fits best in queue will executed first.

vi **Priority Scheduling Algorithm:** This algorithm works on basis of priorities set for each tasks. It considers highest priority task for execution first. Wherever same priority task is found, then it is executed according to FCFS manner. Priority of task is increased with the increase in time. It is very easy to implement. It takes less time for execution. It is good for application, which requires resources. However, low priority task will be lost if system crashes. Starvation for resources.

vii **Shortest Job First Scheduling Algorithm:** This algorithm will find shortest task and is assigned for execution. Task with large execution time has to wait until short task completes its execution(Jadhav, n.d.).

b. **Online Mode Heuristic Algorithms (OHMA):** Cloud environment is heterogeneous and speed of every processor is different from each other. This type of scheduling is more suitable for cloud computing environment.

Scheduling:

Cloud task scheduling is divided into three parts (Malik et al., 2018):

a. Resource searching and filtering: First cloud service provider finds list of available resources from specific network and then checks for their status.
b. Selection of resources: This stage is also known as deciding stage. According to request of a particular resource will be chosen based on their parameters and requirements.
c. Submission of task: After resource selection, the task is assign to resource for executing.

3. Related Work

(Al-Bakhrani et al., 2020) presents the services provided by the simulation program, which can be used in the educational aspect as it facilitates the students to understand the scheduling algorithms within the CPU.

(Shukri et al., 2021)have done comparison analysis with Multi Verse Optimization (MVO) and Particle Swarm Optimization (PSO) with EMVO in terms of achieving minimized makespan and increasing resources utilization.

(Hayatunnufus et al., 2020) provided scheduling on specific device like IOT. Using such implementation it minimizes time consumption for response time and waiting time for each process using Round Robin and FIFO scheduling in the Raspbian and Arch Linux operating systems.

(Sanaj & Prathap, 2020) uses cloudsim simulator for visualizing and understanding complete process. The task features are extracted from the client's task. Then, the features are reduced by using the MRQFLDA algorithm. After that, the large tasks are separated into sub-tasks using a map-reduce framework. Finally, the tasks are efficiently scheduled by using the GA-WOA algorithm. The experimental simulations are carried out using the cloudsim environment.

(Ashish et al., 2020) considered all basic types of scheduling algorithms and made the comparison based on different factors of its criteria. They showed all merits and demerits of every algorithms.

(Alworafi et al., 2019) had designed hybrid SJF-LJF for scheduling task on cloud computing environment. Quality of Service mainly concerns minimizing the total completion time of tasks (i.e., makespan), response time, and increasing the efficiency of resource utilization.

(Garg et al., 2019) created their own algorithm for generation of energy efficient task scheduling using Reliability and Energy Efficient Workflow Scheduling (REEWS) in heterogeneous cloud systems with dynamic voltage and frequency scaling.

(Sharma et al., 2019) made a strategy for both application tasks (depending on data location and availability) and operating system services. The cost of reconfiguration in both tasks and service placement were evaluated and would serve as a metric for the scheduler.

(Panwar et al., 2019)worked on two phase scheduling. In first phase, TOPSIS algorithm is applied to obtain the relative closeness of tasks with respect to selected scheduling criteria (i.e., execution time, transmission time and cost). In second phase, the particle swarm optimization (PSO) begins with computing relative closeness of the given three criteria for all tasks in all VMs. There was improvement of execution time, MakeSpan, resource utilization, processing cost, and transmission time as compared to the dynamic scheduling algorithms.

(Malik et al., 2018)compared various scheduling algorithms to increase the efficiency and performance of cloud environment by finding the best algorithm.

(Jadhav, n.d.) Proved that SJF is better by comparitive analysis with SJF, RR, FCFS, Priority based scheduling algorithms.

(Dubey et al., 2018)evaluated by comparing various algorithm and proved that efficiency of updated Heterogeneous Earliest Final Time is better than original one.

(Fataniya & Patel, 2018)defined Dynamic Time Quantum for RR and proved that Dynnamic RR is better than MRRA.

(Neelima & Reddy, 2018) worked to overcome the drawbacks of SLGSA and FA by combining both and generated a hybrid self-adaptive learning global search algorithm and firefly algorithm (HSLGSAFA)-based model for task scheduling in cloud computing.

(Sujith Kumar Reddy et al., 2018)designed Check Mean round robin (CMRR) algorithm using a mean technique to improve the round robin (RR) algorithm to get optimized waiting and turnaround time with less number of context switching for the given process.

(Pradeep & Jacob, 2018)worked on a hybridization of cuckoo search and gravitational search algorithm (CGSA) to execute the entire task with low cost, less resource use, and less energy consumption.

(Devipriya & Ramesh, 2013) identified that instead of applying larger tasks to faster resources it is applied to smaller task and vice a versa in updated version of Min-Max algorithm implementation for less makespan.

(S. C. Wang et al., 2011) worked on the combination of OLB and MM and identified that it enhances the system performance about 50% while the combination of EOLB and MM enhances performance about 20%.

(L. Wang et al., 2008) reviewed recent advances of cloud computing and presents our early definition of Cloud computing, its interfaces, and its features.

(Singh et al., 2017) studied comparison between different scheduling algorithms in grid and cloud environment by considering different parameters like mode of execution, scheduling factors, scheduling parameter.

Table.1. Comparison of different Scheduling algorithms in Cloud Environment(Singh et al., 2017)

	Decision Mode	**Throughput**	**Response Time**	**Overhead**	**Waiting time**	**Turnaround time**	**Disadvantages**
FCFS	Non-preemptive	Not specific	High	Minimum	Low	Low	Long waiting time, favored CPU over I//O processing, lower device utilization, Incompatibility with time sharing systems.
RR	Preemptive	Depends on quantum	Can be high	Minimum	High	High	Low slicing time reduces processor output, More context switching, Performance depends on time quantum, Processes do not have priorities, No special priority to more important tasks.
Min-Min	Preemptive	High	Moderate	Can be high	Moderate	Moderate	A high value of makespan being generated and low resource utilization.
Max-Min	Preemptive	High	Moderate	Can be high	Moderate	Moderate	Tasks have high waiting time, Not give any priority to tasks, Resources are not consumed in an optimal manner.
Most Fit	Preemptive	High	Moderate	Can be high	Moderate	Moderate	It is a Slow Process.
Priority	Preemptive	Not specific	High	Can be high	High	High	Lose all the low-priority processes if the system crashes, Causes starvation if high-priority processes take

							too much CPU time, The lower priority process can also be postponed for an indefinite time.
SJF	Preemptive	High	Moderate	Can be high	Low	Low	Long turnaround times or starvation, Knowledge about the runtime length of a process is necessary.

4. Conclusion

As cloud environments are designed for availability of services, within the given time, selecting appropriate task scheduling algorithm is very important which minimizes overload on service provider, provides high resource utilization, VM should not be idle, is energy efficient with quick response time and less waiting time. A short description of some scheduling algorithms are given by considering some factors like overload, response time, turnaround time, waiting time, etc(Garg et al., 2019). By improving some factors and by making some changes in existing algorithms, efficiency can be improved in cloud environment.

References

Al-Bakhrani, A. A., Hagar, A. A., Hamoud, A. A., & Kawathekar, S. (2020). Comparative analysis of cpu scheduling algorithms: Simulation and its applications. *International Journal of Advanced Science and Technology*, *29*(3), 483–494.

Alworafi, M. A., Dhari, A., El-Booz, S. A., Nasr, A. A., Arpitha, A., & Mallappa, S. (2019). An enhanced task scheduling in cloud computing based on hybrid approach. In *Lecture Notes in Networks and Systems* (Vol. 43). Springer Singapore. https://doi.org/10.1007/978-981-13-2514-4_2

Ashish, Rauthan, M. M. S., Barthwal, V., & Varma, R. (2020). The latest review: Scheduling algorithms on the cloud computing based environment. *International Journal of Scientific and Technology Research*, *9*(1), 2387–2391.

Devipriya, S., & Ramesh, C. (2013). Improved max-min heuristic model for task scheduling in cloud. *Proceedings of the 2013 International Conference on Green Computing, Communication and Conservation of Energy, ICGCE 2013*, 883–888. https://doi.org/10.1109/ICGCE.2013.6823559

Dubey, K., Kumar, M., & Sharma, S. C. (2018). Modified HEFT Algorithm for Task Scheduling in Cloud Environment. *Procedia Computer Science*, *125*, 725–732. https://doi.org/10.1016/j.procs.2017.12.093

Fataniya, B., & Patel, M. (2018). Dynamic Time Quantum Approach to Improve Round Robin Scheduling Algorithm in Cloud Environment. *Ijsrset*, *4*(4), 963–969. www.ijsrset.com

Garg, R., Mittal, M., & Son, L. H. (2019). Reliability and energy efficient workflow scheduling in cloud environment. *Cluster Computing*, *22*(4), 1283–1297. https://doi.org/10.1007/s10586-019-02911-7

Hayatunnufus, Riasetiawan, M., & Ashari, A. (2020). Performance Analysis of FIFO and Round Robin Scheduling Process Algorithm in IoT Operating System for Collecting Landslide Data. *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics, DATABIA 2020 - Proceedings*, 63–68. https://doi.org/10.1109/DATABIA50434.2020.9190608

Jadhav, P. D. Y. (n.d.). *Analytical study of cpu scheduling algorithms*.

Malik, B. H., Amir, M., Mazhar, B., Ali, S., Jalil, R., & Khalid, J. (2018). Comparison of task scheduling algorithms in cloud environment. *International Journal of Advanced Computer Science and Applications*, *9*(5), 384–390. https://doi.org/10.14569/IJACSA.2018.090550

Neelima, P., & Reddy, A. R. M. (2018). An Efficient Hybridization Algorithm Based Task Scheduling in Cloud Environment. *Journal of Circuits, Systems and Computers*, *27*(2), 1–25. https://doi.org/10.1142/S0218126618500184

Panwar, N., Negi, S., Rauthan, M. M. S., & Vaisla, K. S. (2019). TOPSIS–PSO inspired non-preemptive tasks scheduling algorithm in cloud environment. *Cluster Computing*, *22*(4), 1379–1396. https://doi.org/10.1007/s10586-019-02915-3

Pradeep, K., & Jacob, T. P. (2018). CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment. *Information Security Journal*, *27*(2), 77–91. https://doi.org/10.1080/19393555.2017.1407848

Sanaj, M. S., & Prathap, P. M. J. (2020). An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment. *Materials Today: Proceedings*, *37*(Part 2), 3199–3208. https://doi.org/10.1016/j.matpr.2020.09.064

Sharma, A., Kesarwani, S., & Kumar, K. (2019). *Scheduling of Operating System Services*. *4*.

Shukri, S. E., Al-Sayyed, R., Hudaib, A., & Mirjalili, S. (2021). Enhanced multi-verse optimizer for task scheduling in cloud computing environments. *Expert Systems with Applications*, *168*, 114230. https://doi.org/10.1016/j.eswa.2020.114230

Singh, A. B., Bhat, S., Raju, R., & Souza, R. D. ’. (2017). A Comparative Study of Various Scheduling Algorithms in Cloud Computing. *American Journal of Intelligent Systems*, *7*(3), 68–72. https://doi.org/10.5923/j.ajis.20170703.06

Sujith Kumar Reddy, N., Santhi, H., Gayathri, P., & Jaisankar, N. (2018). A new CPU scheduling algorithm using round-robin and mean of the processes. In *Advances in Intelligent Systems and Computing* (Vol. 732). Springer Singapore. https://doi.org/10.1007/978-981-10-8533-8_23

Wang, L., Tao, J., Kunze, M., Castellanos, A. C., Kramer, D., & Karl, W. (2008). Scientific cloud computing: Early definition and experience. *Proceedings - 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008*, 825–830. https://doi.org/10.1109/HPCC.2008.38

Wang, S. C., Yan, K. Q., Wang, S. S., & Chen, C. W. (2011). A three-phases scheduling in a hierarchical cloud computing network. *Proceedings - 2011 3rd International Conference on Communications and Mobile Computing, CMC 2011*, 114–117. https://doi.org/10.1109/CMC.2011.28