International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

IJIEMR Transactions, online available on 11th Sept 2023. Link

:http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 09

10.48047/IJIEMR/V12/ISSUE 09/32

Title **Various Task Scheduling Algorithms in Cloud**

Volume 12, ISSUE 09, Pages: 263-270

Paper Authors **Dr.Sateesh Nagavarapu,N.Pavan,Ch.Udaya Bhanu,Varshitha Reddy.B**

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Various Task Scheduling Algorithms in Cloud

[1] Dr.Sateesh Nagavarapu,[2] N.Pavan,[3] Ch.Udaya Bhanu,[4] Varshitha Reddy.B
[1]Professor, [2]Assistant Professor, [3,4]Student,
[1,2,3,4] Computer Science & Engineering
[1,2,3,4] Malla Reddy Institute of Technology, Maisammaguda, Hyderabad-500100
[1]sateeshnagavarapu@gmail.com,
[2]npavan26@gmail.com
[3]bhanu17ub@gmail.com
[4]varshithareddy761 @gmail.com

Abstract:

Cloud computing almost trending technologies in now-a-day, which enables users (including individuals and businesses) to remotely access computer resources (software, hardware, and platform) as services over the Internet. Scalability, changeable costs, accessibility, dependability, and on-demand pay-as-you-go services set cloud computing apart from conventional computing paradigms. Cloud computing must be able to answer all user demands with high performance and a guaranty of service quality because it serves millions of users concurrently (QoS). Hence, in order to fairly and effectively fulfill these requirements, we must build a suitable job scheduling system. One of the most important problems in the cloud computing environment is task scheduling because it directly affects how well the cloud performs.There are many different kinds of scheduling algorithms, including dynamic scheduling algorithms that are thought to be appropriate for large-scale cloud computing environments and static scheduling methods that are thought to be ideal for small or medium-sized cloud computing. In this study, we aim to demonstrate the performance of the three most widely used static task scheduling algorithms: first come first served (FCFS), short job first scheduling (SJF), and MAX-MIN. Their effects on algorithm complexity, resource availability, total execution time (TET), total waiting time (TWT), and total finish time(TFT) have been evaluated using the CloudSim simulator.

Keywords: task scheduling algorithms, load balance, performance

Introduction:

Using computer resources (hardware, software, and platforms) as a service and making them available to customers on demand over the Internet is known as cloud computing, a novel technology that evolved from grid computing and distributed computing [1]. It is the first technological advancement to use the idea of commercialized computer science deployment with general users [2]. Using the virtualization concept, it depends on consumers resource sharing. Cloud computing may deliver superior efficiency via evenly and efficiently spreading workloads across all resources to reduce waiting times, increase throughput, and manage resources to their fullest potential.

Even so, task scheduling and load balancing are the largest since they are thought to be the primary elements that affect other performance metrics including availability, scalability, and power consumption.

2. Tasks scheduling algorithms overview

Algorithms for scheduling tasks are described as the process for choosing the resources to carry out activities with the least amount of waiting and execution time.

2.1 Scheduling levels

Two levels of scheduling algorithms exist in the cloud computing environment:

- **First level:** A set of policies are present at the host level, to distribute VMs.
- **Second level:** A collection of policies to assign tasks to VMs are found at the VM level.

We chose task scheduling algorithms as a research topic because it is the greatest challenge in cloud computing and the primary factor that controls the performance standards, such as (execution time, response time, waiting time, network, bandwidth, and service cost) for all tasks, as well as controlling other factors that can affect performance, such as power usage, accessibility, expandability, storage capabilities, and buffer cap.

2.2 Tasks scheduling algorithms definition and advantages

A group of rules and policies known as tasks scheduling algorithms are used to allocate tasks to the appropriate resources (CPU, memory, and bandwidth) in order to achieve the best level of efficiency and resource utilization.

2.2.1 Task scheduling algorithms advantages

- Control QoS and performance in the cloud.
- Control the Processor and RAM.
- Effective scheduling algorithms reduce overall job execution time while maximizing resource use.
- Ensuring more equity across all tasks.
- Raising the quantity of tasks that are successfully finished.
- Using a real-time system to schedule tasks.
- Obtaining a high throughput for the system.
- Better load distribution.

2.3 Tasks scheduling algorithms classifications

Tasks scheduling algorithms classified as in **Figure 1**.

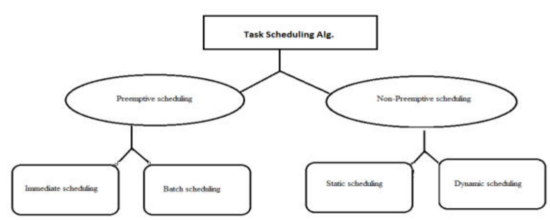

Figure 1. *Tasks scheduling classes.*

2.3.1 Tasks scheduling algorithms can be classified as follows

- Immediate scheduling: New tasks are immediately scheduled to VMs as they occur.
- Batch scheduling: Prior to sending, tasks are compiled into a batch; this type is sometimes referred to as mapping events.
- Static scheduling: When opposed to dynamic scheduling, static scheduling is seen to be fairly straightforward; it is

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

predicated on previous knowledge about the system's overall state. It allocates all traffic equally among all VMs in a similar way to round robin (RR) and random scheduling algorithms, without taking into account the status of the VMs.

- Dynamic scheduling: distributes the tasks according to the capacity of all available VMs while taking into account the present condition of the virtual machines and does not require prior knowledge of the system's overall state [4-6].
- Preemptive scheduling: Each task's execution is paused and can be transferred to another resource to finish [6].
- Non-preemptive scheduling: Virtual machines (VMs) are not reallocated to new tasks until the scheduled job has completed execution [6].

We concentrate on static scheduling techniques in this study. First come first served (FCFS), shortest job first (SJF), and MAXMAX scheduling algorithms are examples of static scheduling algorithms that are simple and inexpensive on a small or medium scale.

2.4 Task scheduling system in cloud computing

The cloud computing job scheduling system has three layers [7].

- The first task level consists of a collection of tasks (called Cloudlets) provided by cloud users and necessary for execution.
- The second scheduling level is in charge of allocating tasks to the appropriate resources to achieve the best resource utilization with the shortest lead time. The make span is the total amount of time required to complete all jobs from start to finish [7].
- A group of virtual machines (VMs) used to carry out tasks such as

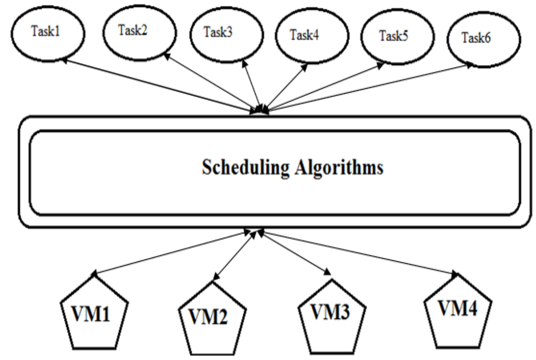

Figure 2. *Task scheduling system.*

2.5 This level passes through two steps

- Using a datacenter broker, the initial stage entails finding and sorting through all of the VMs that are currently present in the system and gathering status data on each of them [8].
- A suitable VM is chosen in the second stage depending on the task properties [8].

3. Static tasks scheduling algorithms in cloud computing environment

3.1 FCFS

FCFS: After being assigned to VMs, the order of the jobs in the task list is determined by their arrival time [3].

3.1.1 Advantages

- Fairer than other simple scheduling algorithms;
- Most widely used and simplest scheduling algorithm.
- Rely on the FIFO rule when allocating tasks.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

- Simpler than alternative scheduling algorithms.

3.1.2 Disadvantages

- Waiting times for tasks are long.
- Not giving any duties any priority. This implies that all tasks must wait a very long time for the huge tasks in the begin tasks list to finish.
- Resources are not used in the best possible way.
- We will test them and then gauge their effect on (fairness, ET, TWT, and TFT) in order to gauge the performance attained by this strategy.

3.1.3 Assumptions

When assigning tasks to VMs in the cloud computing environment, some presumptions must be considered.

- There should be more tasks than virtual machines, thus each VM must carry out many duties.
- There is only one VM resource per process.
- Tasks of various sizes, including small, medium, and large.
- Tasks aren't stopped mid-execution once they've begun.
- The available VMs are of exclusive use and cannot be shared among various jobs.
- VMs are independent in terms of resources and control. It implies that the VMs won't be able to contemplate taking on new duties until the current tasks have been completed [3].
- Assume there are 15 tasks, each with a length as listed in Table 1.

TASK	LENGTH
T1	100002
T2	70000
T3	5000
T4	1000
T5	3000
T6	6000
T7	100000
T8	15000
T9	7000
T10	4000
T11	3000
T12	8000
T13	100000
T14	25000
T15	20000

Table 1.*Set of tasks with different length orders depends on the arrival time for each task.*

3.1.4 VM properties

Assuming we have six virtual machines (VMs) with various characteristics depending on the magnitude of the task:

VM1 through VM6 are on the VM list.
MIPS of the VM list are 500, 500, 1500, 1500, and 2500, respectively.

To optimize the load balance, we chose a group of VMs with a variety of characteristics so that each category would include VMs that were capable of handling a particular class of activities. Because one class differs from the others in terms of work lengths, load imbalance results when we utilize VMs with the same features across all categories.

3.1.5 When applying FCFS, work mechanism will be as following

Figure 3 depicts the operation of the FCFS tasks scheduling algorithm and how tasks are carried out in accordance with their arrival time. The initial sets of duties are

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

indicated by dot arrows and are scheduled according to arrival time. The second set of jobs are scheduled based on their arrival time and are indicated by dash arrows. Third group of jobs that are scheduled according to arrival time are shown by solid arrows. We can see from this that t1 is excessively large when compared to t7 and t12. TWT, ET, TFT, and fairness all rise as a result of the fact that t7 and t12 must wait for t1.

VM1 = {t1 ⟶ t7 ⟶ t12} .
VM2 = {t2 ⟶ t8 ⟶ t14} .
VM3 = {t3 ⟶ t9 ⟶ t15} .
VM4 = {t4 ⟶ t10} .
VM5 = {t5 ⟶ t11} .
VM6 = {t6 ⟶ t13} .

Table 2 shows how FCFS scheduling algorithm increases waiting time for all tasks.

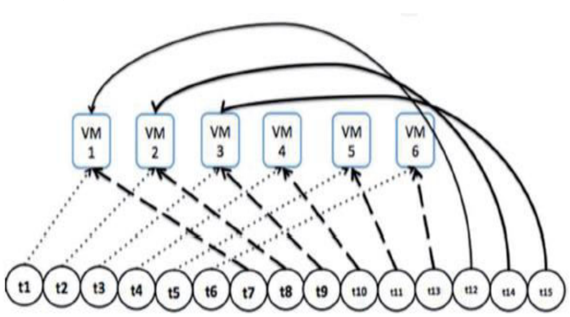

Figure 3.*FCFS work mechanism.*

Task	ET	Waiting Time		
T1	200	VM1		
T2	140	VM2		
T3	3.33	VM3		
T4	0.66	VM4		
T5	1.2	VM5		
T6	4	VM6		
T7	180	Wait(200)	VM1	
T8	200	Wait(140)	VM2	
T9	10	Wait(3.33)	VM3	
T10	0.66	Wait(0.66)	VM4	
T11	0.8	Wait(1.2)	VM5	
T12	1.6	Wait(4)	VM6	
T13	40	Wait(380)		VM1
T14	50	Wait(340)		VM2
T15	53.33	Wait(13.33)		VM3

Table2. *Waiting times of tasks in FCFS.*

3.2 SJF

According to their importance, tasks are sorted. According to task duration, tasks are ranked in order of highest priority to lowest task.

3.2.1 Advantages

- Less wait time than FCFS.
- The SJF job scheduling method has the lowest average waiting time.

3.2.2 Disadvantages

- When tasks are sent to VM, some tasks are treated unfairly since large jobs frequently get pushed to the bottom of the task list while short tasks are given to VM.
- Takes a lot of TFT and execution time

3.2.3 SJF work mechanism

The following work mechanism will be used when applying SJF:

Suppose that Table 1's 15 tasks are accurate. The job list will be sorted as shown in Table 3. According to their lengths, as shown in Table 3, tasks are sorted from least to largest before being sequentially assigned to virtual machines.

3.2.4 Execute tasks will be

VM1 = {t4 ⟶ t6 ⟶ t7} .
VM2 = {t10 ⟶ t9 ⟶ t1} .
VM3 = {t11 ⟶ t13 ⟶ t8} .
VM4 = {t5 ⟶ t14} .
VM5 = {t12 ⟶ t2} .
VM6 = {t3 ⟶ t15} .

Table 4 demonstrates that the big chores must wait in the task list until the little ones are done.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

3.3 MAX-MIN

With MAX-MIN, jobs are prioritized according to their length; lengthy tasks take longer to complete than shorter chores. Then it was given to the VM in the list of VMs with the shortest overall execution duration.

3.3.1. Advantages

This algorithm outperforms the FCFS, SJF, and MIN-MIN algorithms by efficiently utilising the resources that are available.

3.3.2 Disadvantages

The MAX-MIN scheduling method will give priority to six lengthy activities in the VM list, so short tasks must wait until the large tasks are finished. This will lengthen the waiting period for tiny and medium tasks.

When assignments are given to VM, there is unfairness to some or most small and medium tasks.

• The Work Mechanism when using MAX-MIN will be as follows.

Task	T4	T5	T11	T10	T3	T6	T9	T12	T8	T15	T14	T2	T7	T13	T1
Length	1000	3000	3000	4000	5000	6000	7000	8000	15000	20000	25000	70000	100000	100000	100002

Table 3. *A set of tasks sorted based on SJF scheduling algorithm.*

Task	ET	Waiting Time		
T4	2	VM1		
T5	2	VM2		
T11	1.33	VM3		
T10	2	VM4		
T3	1.6	VM5		
T6	2	VM6		
T9	20	Wait(2)	VM1	
T12	30	Wait(2)	VM2	
T8	1.33	Wait(1.33)	VM3	
T15	16.66	Wait(2)	VM4	
T14	28	Wait(1.6)	VM5	
T2	32	Wait(2)	VM6	
T7	180	Wait(22)		VM1
T13	200	Wait(32)		VM2
T1	66.66	Wait(14.66)		VM3

Table 4. *Waiting times of tasks in SJF.*

Task	T1	T7	T13	T2	T14	T15	T8	T12	T9	T10	T11	T10	T11	T5	T4
Length	100002	100000	100000	70000	25000	20000	15000	8000	7000	4000	3000	4000	3000	3000	1000

Table 5. *A set of tasks sorted based on MAX-MIN scheduling algorithm.*

Suppose that Table 1's 15 tasks are accurate. As shown in Table 5, we shall order the tasks in the task list. Based on highest completion time, tasks are arranged from greatest to smallest. These are then allocated to the VMs in the list of VMs with the shortest total execution time.

3.3.3 Execute tasks will be

$$VM6 = \{t1 \longrightarrow t13 \longrightarrow t11\}\ .$$
$$VM5 = \{t8 \longrightarrow t9 \longrightarrow t10\}\ .$$
$$VM4 = \{t7 \longrightarrow t6 \longrightarrow t4\}\ .$$
$$VM3 = \{t15 \longrightarrow t3\}\ .$$
$$VM2 = \{t2 \longrightarrow t12\}\ .$$
$$VM1 = \{t14 \longrightarrow t5\}\ .$$

According to Tables 6 and 7, the small and medium tasks must wait on the task list until the major tasks are finished being completed.

The TWT and TFT for the three task scheduling algorithms FCFS, SJF, and MAX-MIN are shown in Figure 4. The best task scheduling method in terms of TWT and TFT is SJF.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Task	ET	Waiting Time		
T1	40	VM6		
T7	40	VM5		
T13	60	VM4		
T2	53.33	VM3		
T14	140	VM2		
T15	50	VM1		
T8	8	Wait(40)	VM6	
T12	6	Wait(40)	VM5	
T9	6.66	Wait(60)	VM4	
T6	3.33	Wait(53.33)	VM3	
T3	8	Wait(140)	VM2	
T10	6	Wait(50)	VM1	
T5	0.8	Wait(48)		VM6
T11	0.4	Wait(46)		VM5
T4	0.67	Wait(66.67)		VM4

Table 6. *Waiting time of tasks in MIX-MIN scheduling algorithm.*

	FCFS	SJF	MAX-MIN
TWT	739.19	79.59	404
TFT	1969.69	678.69	968.698

Table 7. *Comparison between FCFS tasks scheduling algorithm, SJF, and MAX-MIN in terms of TWT and TFT.*

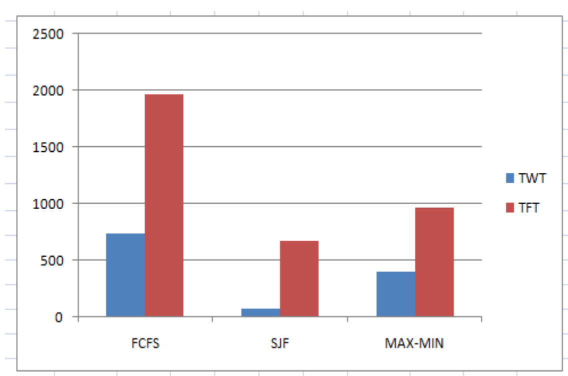

Figure.4. *Comparison between FCFS tasks scheduling algorithm, SJF, and MAX-MIN in terms of TWT and TFT.*

4. Conclusion

This chapter introduces the meaning of the tasks scheduling algorithms and types of static and dynamic scheduling algorithms in cloud computing environment. This chapter also introduces a comparative study between the static task scheduling algorithms in a cloud computing environment such as FCFS, SJF, and MAX-MIN, in terms of TWT, TFT, fairness between tasks, and when becoming suitable to use?

Experimentation was executed on CloudSim, which is used for modeling the different tasks scheduling algorithms.

5.References

[1] Ramotra A, Bala A. Task-Aware Priority Based Scheduling in Cloud Computing [master thesis]. Thapar University; 2013

[2] Microsoft Azure website. [Accessed:01 October 2017]

[3] Kumar Garg S, Buyya R. Green Cloud Computing and Environmental Sustainability, Australia: Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering, The University of Melbourne; 2012

[4] Al-maamari A, Omara F. Task scheduling using PSO algorithm in cloud computing environments. International Journal of Grid Distribution Computing. 2015;8(5):245-256

[5] http://www.pbenson.net/2013/04/thecloud-defined-part-1-of-8-on-demandself-service/ [Accessed: 01 October 2017]

[6] Endo P, Rodrigues M, Gonçalves G, Kelner J, Sadok D, Curescu C. High availability in clouds: Systematic review and research challenges. Journal of Cloud Computing Advances, Systems and Applications. 2016

[7] http://www.techinmind.com/ what-is-cloud-computing-what-areits-advantages-and-disadvantages/ [Accessed: 01 October 2017]

[8] https://siliconangle.com/ blog/2016/04/29/survey-sees-rapid growth-in-enterprise-cloud-adoption/[Accessed: 01 October 2017]