Chapter

Types of Task Scheduling Algorithms in Cloud Computing Environment

Tahani Aladwani

Abstract

Cloud computing is one of the most important technologies used in recent times, it allows users (individuals and organizations) to access computing resources (software, hardware, and platform) as services remotely through the Internet. Cloud computing is distinguished from traditional computing paradigms by its scalability, adjustable costs, accessibility, reliability, and on-demand pay-as-you-go services. As cloud computing is serving millions of users simultaneously, it must have the ability to meet all users requests with high performance and guarantee of quality of service (QoS). Therefore, we need to implement an appropriate task scheduling algorithm to fairly and efficiently meet these requests. Task scheduling problem is the one of the most critical issues in cloud computing environment because cloud performance depends mainly on it. There are various types of scheduling algorithms; some of them are static scheduling algorithms that are considered suitable for small or medium scale cloud computing; and dynamic scheduling algorithms that are considered suitable for large scale cloud computing environments. In this research, we attempt to show the most popular three static task scheduling algorithms performance there are: first come first service (FCFS), short job first scheduling (SJF), MAX-MIN. The CloudSim simulator has been used to measure their impact on algorithm complexity, resource availability, total execution time (TET), total waiting time (TWT), and total finish time (TFT).

Keywords: task scheduling algorithms, load balance, performance

1. Introduction

Cloud computing is a new technology derived from grid computing and distributed computing and refers to using computing resources (hardware, software, and platforms) as a service and provided to beneficiaries on demand through the Internet [1]. It is the first technology that uses the concept of commercial implementation of computer science with public users [2]. It relies on sharing resources among users through the use of the virtualization technique. High performance can be provided by a cloud computing, based on distributing workloads across all resources fairly and effectively to get less waiting time, execution time, maximum throughput, and exploitation of resources effectively. Still, there are many challenges prevalent in cloud computing, Task scheduling and load balance are the

biggest yet because it is considered the main factors that control other performance criteria such as availability, scalability, and power consumption.

2. Tasks scheduling algorithms overview

Tasks scheduling algorithms are defined as the mechanism used to select the resources to execute tasks to get less waiting and execution time.

2.1 Scheduling levels

In the cloud computing environment there are two levels of scheduling algorithms:

- First level: in host level where a set of policies to distribute VMs in host.
- Second level: in VM level where a set of policies to distribute tasks to VM.

In this research we focus on VM level to scheduling tasks, we selected task scheduling algorithms as a research field because it is the biggest challenge in cloud computing and the main factor that controls the performance criteria such as (execution time, response time, waiting time, network, bandwidth, services cost) for all tasks and controlling other factors that can affect performance such as power consumption, availability, scalability, storage capacity, buffer capacity, disk capacity, and number of users.

2.2 Tasks scheduling algorithms definition and advantages

Tasks scheduling algorithms are defined as a set of rules and policies used to assign tasks to the suitable resources (CPU, memory, and bandwidth) to get the highest level possible of performance and resources utilization.

2.2.1 Task scheduling algorithms advantages

- Manage cloud computing performance and QoS.
- Manage the memory and CPU.
- The good scheduling algorithms maximizing resources utilization while minimizing the total task execution time.
- Improving fairness for all tasks.
- Increasing the number of successfully completed tasks.
- Scheduling tasks on a real-time system.
- Achieving a high system throughput.
- Improving load balance.

2.3 Tasks scheduling algorithms classifications

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

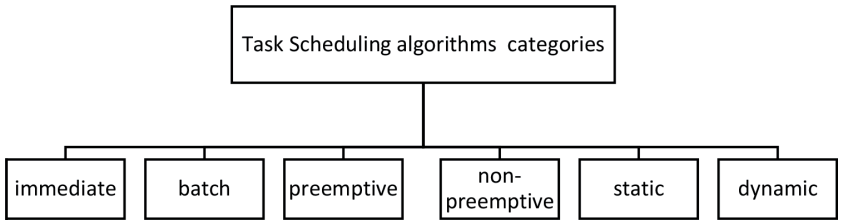

Figure 1.
Tasks scheduling classes.

2.3.1 Tasks scheduling algorithms can be classified as follows

- Immediate scheduling: when new tasks arrive, they are scheduled to VMs directly.

- Batch scheduling: tasks are grouped into a batch before being sent; this type is also called mapping events.

- Static scheduling: is considered very simple compared to dynamic scheduling; it is based on prior information of the global state of the system. It does not take into account the current state of VMs and then divides all traffic equivalently among all VMs in a similar manner such as round robin (RR) and random scheduling algorithms.

- Dynamic scheduling: takes into account the current state of VMs and does not require prior information of the global state of the system and distribute the tasks according to the capacity of all available VMs [4–6].

- Preemptive scheduling: each task is interrupted during execution and can be moved to another resource to complete execution [6].

- Non-preemptive scheduling: VMs are not re-allocated to new tasks until finishing execution of the scheduled task [6].

In this research, we focus on the static scheduling algorithms. Static scheduling algorithm such as first come first service (FCFS), shortest job first (SJF), and MAX-MAX scheduling algorithms in complexity and cost within a small or medium scale.

2.4 Task scheduling system in cloud computing

The task scheduling system in cloud computing passes through three levels [7].

- The first task level: is a set of tasks (Cloudlets) that is sent by cloud users, which are required for execution.

- The second scheduling level: is responsible for mapping tasks to suitable resources to get highest resource utilization with minimum makespan. The makespan is the overall completion time for all tasks from the beginning to the end [7].

- The third VMs level: is a set of (VMs) which are used to execute the tasks as in **Figure 2**.

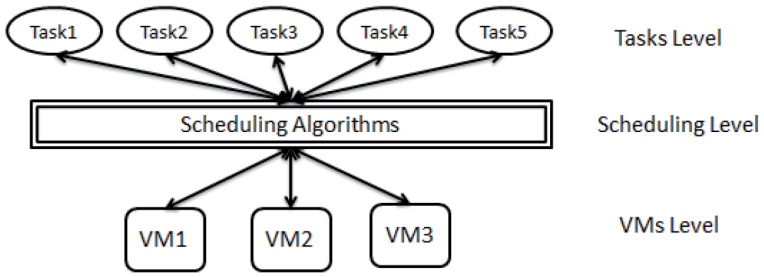

Figure 2.
Task scheduling system.

2.5 This level passes through two steps

- The first step is discovering and filtering all the VMs that are presented in the system and collecting status information related to them by using a datacenter broker [8].
- In the second step a suitable VM is selected based on task properties [8].

3. Static tasks scheduling algorithms in cloud computing environment

3.1 FCFS

FCFS: the order of tasks in task list is based on their arriving time then assigned to VMs [3].

3.1.1 Advantages

- Most popular and simplest scheduling algorithm.
- Fairer than other simple scheduling algorithms.
- Depend on FIFO rule in scheduling task.
- Less complexity than other scheduling algorithms.

3.1.2 Disadvantages

- Tasks have high waiting time.
- Not give any priority to tasks. That means when we have large tasks in the begin tasks list, all tasks must wait a long time until the large tasks to finish.
- Resources are not consumed in an optimal manner.
- In order to measure the performance achieved by this method, we will be testing them and then measuring its impact on (fairness, ET, TWT, and TFT).

3.1.3 Assumptions

Some of the assumptions must be taken into account when scheduling tasks to VMs in the cloud computing environment.

- Number of tasks should be more than the number of VMs, which means that each VM must execute more than one task.
- Each task is assigned to only one VM resource.
- Lengths of tasks varying from small, medium, and large.
- Tasks are not interrupted once their executions start.
- VMs are independent in terms of resources and control.
- The available VMs are of exclusive usage and cannot be shared among different tasks. It means that the VMs cannot consider other tasks until the completion of the current tasks is in progress [3].

Tasks lengths: assume we have 15 tasks with their lengths as in **Table 1**.

Task	Length
t1	100000
t2	70000
t3	5000
t4	1000
t5	3000
t6	10000
t7	90000
t8	100000
t9	15000
t10	1000
t11	2000
t12	4000
t13	20000
t14	25000
t15	80000

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

3.1.4 VM properties

Assume we have six VMs with different properties based on tasks size:

$$\text{VM list} = \{VM1, VM2, VM3, VM4, VM5, VM6\}.$$

$$\text{MIPS of VM list} = \{500, 500, 1500, 1500, 2500, 2500\}.$$

We selected a set of VMs with different properties to make each category have VMs with appropriate ability to serve a specific class of tasks, to improve the load balance. Because when we use VMs with same properties with all categories it leads to load imbalance, where each class is different from other classes in terms of tasks lengths.

3.1.5 When applying FCFS, work mechanism will be as following

Figure 3 shows FCFS tasks scheduling algorithm working mechanism and how tasks are executed based on their arrival time.

Dot arrows refer to first set of tasks scheduling based on their arrival time.

Dash arrows refer to second set of tasks scheduling based on their arrival time.

Solid arrows refer to third set of tasks scheduling based on their arrival time.

And here it is clear to us that t1 is too large compared with t7 and t12. However, t7 and t12 must wait for t1, which leads to an increase in the TWT, ET, TFT, and a decrease in fairness.

$$VM1 = \{t1 \longrightarrow t7 \longrightarrow t12\}.$$

$$VM2 = \{t2 \longrightarrow t8 \longrightarrow t14\}.$$

$$VM3 = \{t3 \longrightarrow t9 \longrightarrow t15\}.$$

$$VM4 = \{t4 \longrightarrow t10\}.$$

$$VM5 = \{t5 \longrightarrow t11\}.$$

$$VM6 = \{t6 \longrightarrow t13\}.$$

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

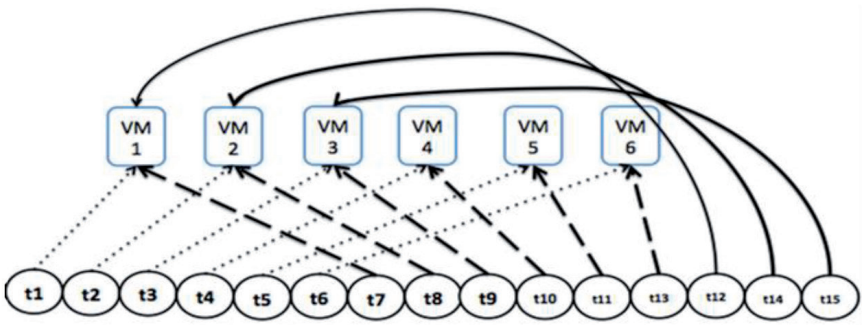

Figure 3.
FCFS work mechanism.

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

Table 2.
Waiting times of tasks in FCFS.

3.2 SJF

Tasks are sorted based on their priority. Priority is given to tasks based on tasks lengths and begins from (smallest task ≡ highest priority).

3.2.1 Advantages

- Wait time is lower than FCFS.
- SJF has minimum average waiting time among all tasks scheduling algorithms.

3.2.2 Disadvantages

- Unfairness to some tasks when tasks are assigned to VM, due to the long tasks tending to be left waiting in the task list while small tasks are assigned to VM.
- Taking long execution time and TFT.

3.2.3 SJF work mechanism

When applying SJF, work mechanism will be as follows:
Assume we have 15 tasks as in **Table 1** above. We will be sorting tasks in the task list, as in **Table 3**. Tasks are sorted from smallest task to largest task based on their lengths as in **Table 3**, then assigned to VMs list sequential.

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 shows that the large tasks must be waiting in the task list until the smallest tasks finish execution.

3.3 MAX-MIN

In MAX-MIN tasks are sorted based on the completion time of tasks; long tasks that take more completion time have the highest priority. Then assigned to the VM with minimum overall execution time in VMs list.

3.3.1. Advantages

- Working to exploit the available resources in an efficient manner.
- This algorithm has better performance than the FCFS, SJF, and MIN-MIN algorithm.

3.3.2 Disadvantages

- Increase waiting time to small and medium tasks; if we have six long tasks, in MAX-MIN scheduling algorithm they will take priority in six VMs in VM list, and short tasks must be waiting until the large tasks finish.

Unfairness to some or most small and medium tasks when tasks are assigned to VM.

- When applying MAX-MIN, Work Mechanism will be as follows.

Tasks	*t4*	*t10*	*t11*	*t5*	*t12*	*t3*	*t6*	*t9*	*t13*	*t14*	*t2*	*t15*	*t7*	*t1*	*t8*
lengths	1000	1000	2000	3000	4000	5000	10000	15000	20000	25000	70000	80000	90000	100000	100000

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

Task	ET	Waiting time		
t4	2	VM1		
t10	2	VM2		
t11	1.33	VM3		
t5	2	VM4		
t12	1.6	VM5		
t3	2	VM6		
t6	20	Wait(2)	VM1	
t9	30	Wait(2)	VM2	
t13	13.33	Wait(1.33)	VM3	
t14	16.66	Wait(2)	VM4	
t2	28	Wait(1.6)	VM5	
t15	32	Wait(2)	VM6	
t7	180	Wait(22)		VM1
t1	200	Wait(32)		VM2
t8	66.66	Wait(14.66)		VM3

Table 4.
Waiting times of tasks in SJF.

Tasks	t1	t8	t7	t15	t2	t14	t13	t9	t6	t3	t12	t5	t11	t10	t4
lengths	100000	100000	90000	80000	70000	25000	20000	15000	10000	5000	4000	3000	2000	1000	1000

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

Assume we have 15 tasks as in **Table 1** above. We will be sorting tasks in task list as in **Table 5**. Tasks sorted from largest task to smallest task based on highest completion time. They are then assigned to the VMs with minimum overall execution time in VMs list.

3.3.3 Execute tasks will be

$$VM6 = \{t1 \longrightarrow t13 \longrightarrow t11\}.$$

$$VM5 = \{t8 \longrightarrow t9 \longrightarrow t10\}.$$

$$VM4 = \{t7 \longrightarrow t6 \longrightarrow t4\}.$$

$$VM3 = \{t15 \longrightarrow t3\}.$$

$$VM2 = \{t2 \longrightarrow t12\}.$$

$$VM1 = \{t14 \longrightarrow t5\}.$$

Tables 6 and 7 shows that the small and medium tasks must be waiting in the task list until the large tasks finish execution.

Figure 4 shows the TWT and TFT for the three tasks scheduling algorithms FCFS, SJF, and MAX-MIN. SJF tasks scheduling algorithm is the best in term of TWT and TFT.

Task	ET	Waiting time		
t1	40	VM6		
t8	40	VM5		
t7	60	VM4		
t15	53.33	VM3		
t2	140	VM2		
t14	50	VM1		
t13	8	Wait(40)	VM6	
t9	6	Wait(40)	VM5	
t6	6.66	Wait(60)	VM4	
t3	3.33	Wait(53.33)	VM3	
t12	8	Wait(140)	VM2	
t5	6	Wait(50)	VM1	
t11	0.8	Wait(48)		VM6
t4	0.4	Wait(46)		VM5
t10	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.

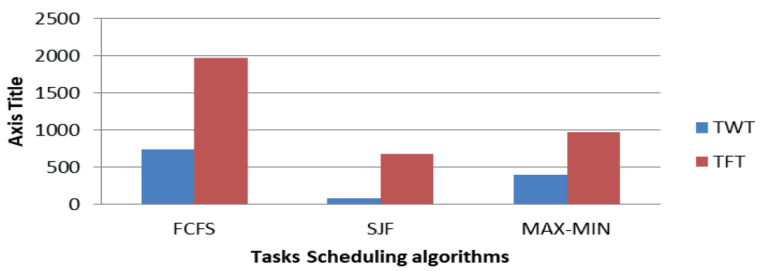

Figure 4.
Comparison between FCFS tasks scheduling algorithm, SJF, and MAX-MIN in terms of TWT and TFT.

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.

DOI: http://dx.doi.org/10.5772/intechopen.86873

Author details

Tahani Aladwani
Mecca, Saudi Arabia

*Address all correspondence to: aladwani_tahani@yahoo.com

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/the-cloud-defined-part-1-of-8-on-demand-self-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-are-its-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]