Scheduling in Cloud Computing

Dr J P Patra

Associate Professor & Head

Dept. of CSE

UTD, CSVTU Bhilai

Definition

The concept of scheduling in cloud computing refers to **the technique of mapping a set of jobs to a set of virtual machines (VMs) or allocating VMs to run on the available resources in order to fulfil users' demands**

Cont.

- Scheduling is a method that is used to distribute valuable computing resources, usually processor time, bandwidth and memory, to the various processes, threads, data flows and applications that need them.

- Scheduling is done to balance the load on the system and ensure equal distribution of resources and give some prioritization according to set rules.

- This ensures that a computer system is able to serve all requests and achieve a certain quality of service.

- Scheduling is also known as process scheduling.

Types

Scheduling in a system is done by the aptly named scheduler, which is mainly concerned with three things:

- **Throughput,** or how fast it can finish a certain number of tasks from beginning to end per unit of time
- **Latency,** which is the turnaround time or the time it takes to finish the task from the time of request or submission until the finish, which includes the waiting time before it could be served
- **Response time**, which is the time it takes for the process or request to be served, in short, the waiting time

Cont.

- Scheduling is largely based on the factors mentioned in the previous slide and varies depending on the system and the programming of the system's or user's preferences and objectives.

- In modern computers such as PCs with large amounts of processing power and other resources and with the ability to multitask by running multiple threads or pipelines at once.

- Scheduling is no longer a big issue and most times processes and applications are given free rein with extra resources, but the scheduler is still hard at work managing requests.

Types of scheduling include:

- First come, first served — The most straightforward approach and may be referred to as first in, first out; it simply does what the name suggests.
- Round robin — Also known as time slicing, since each task is given a certain amount of time to use resources. This is still on a first-come-first-served basis.
- Shortest remaining time first — The task which needs the least amount of time to finish is given priority.
- Priority — Tasks are assigned priorities and are served depending on that priority. This can lead to the starvation of the least important tasks as they are always preempted by more important ones.

Scheduling levels

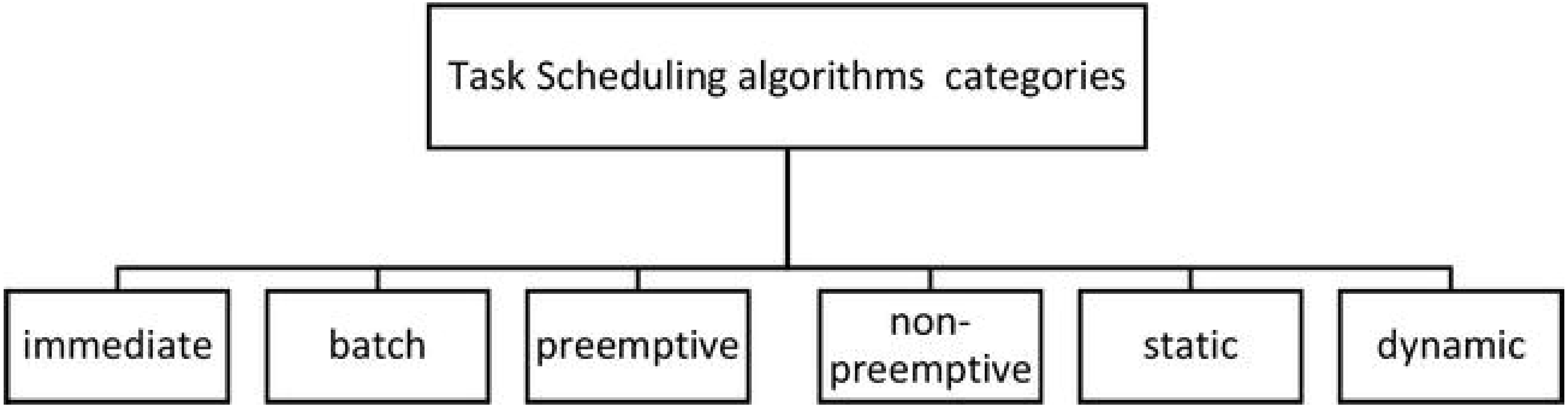
In the cloud computing environment there are two levels of scheduling algorithms:

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

Task scheduling algorithms advantages

- **Manage cloud computing performance**
- **Manage the memory and CPU.**
- **Good scheduling algorithms maximize resource 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.**

Tasks scheduling algorithms classifications



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.

Cont.

- Dynamic scheduling: takes into account the current state of VMs and does not require prior information on the global state of the system and distributes the tasks according to the capacity of all available VMs.

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

- Non-preemptive scheduling: VMs are not reallocated to new tasks until finishing the execution of the scheduled task.

Task scheduling system in cloud computing

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

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

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

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

Scheduling

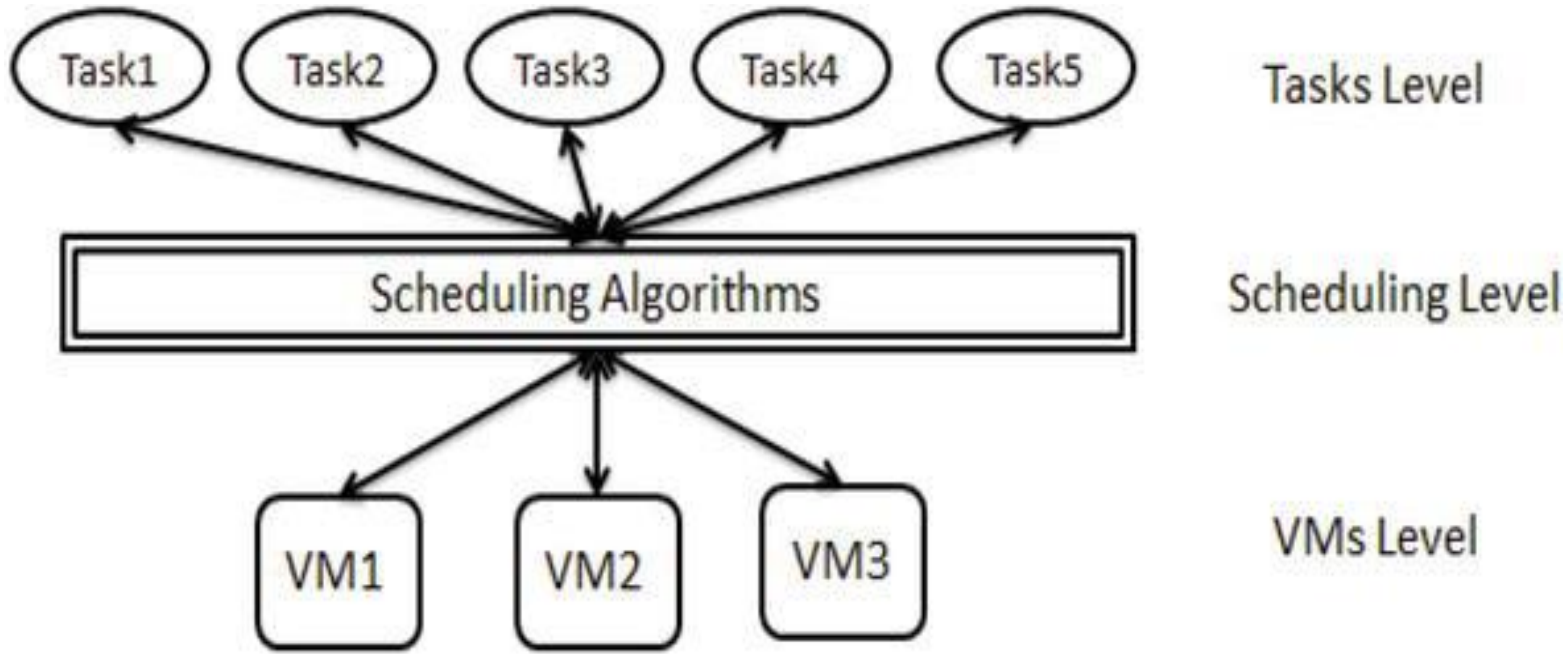


Figure 2

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.

- In the second step a suitable VM is selected based on task properties.

Static tasks scheduling algorithms in cloud computing environment

FCFS: the order of tasks in the task list is based on their arrival time and then assigned to VMs.

Advantages:

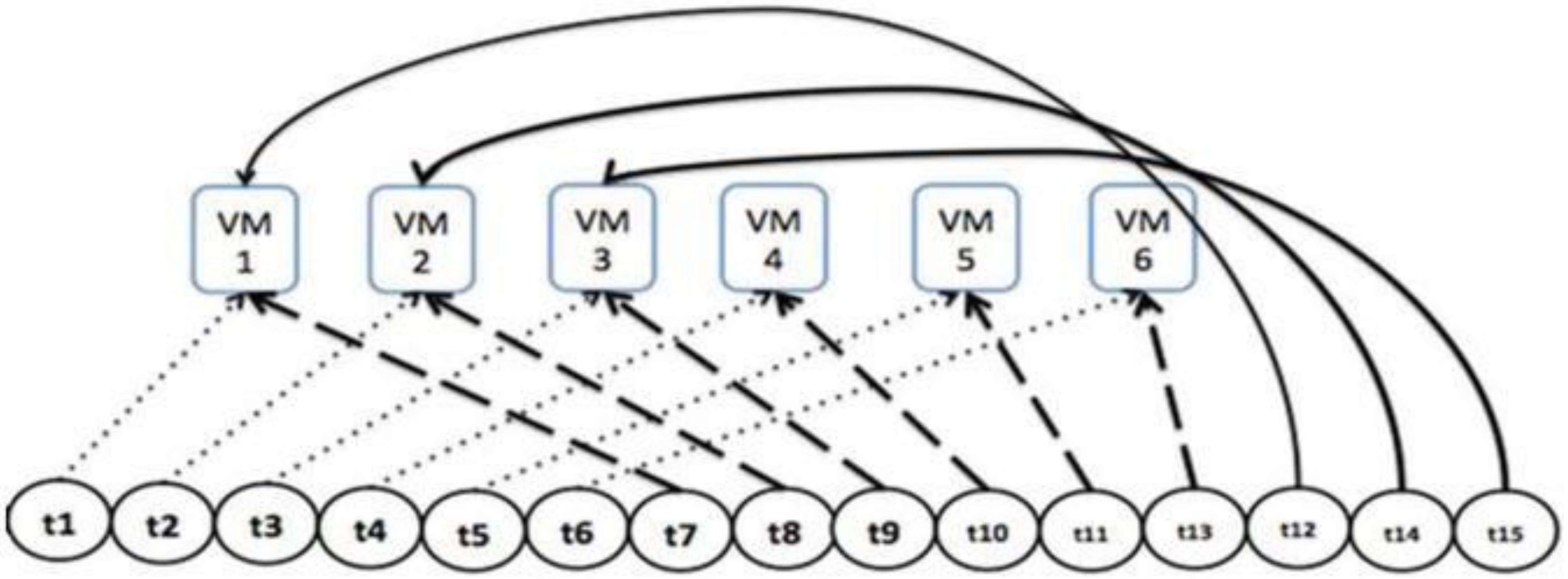
- Most popular and simplest scheduling algorithm.
- Fairer than other simple scheduling algorithms.
- Depend on the FIFO rule in scheduling tasks.
- Less complexity than other scheduling algorithms.

Cont.

Disadvantages of FCFS

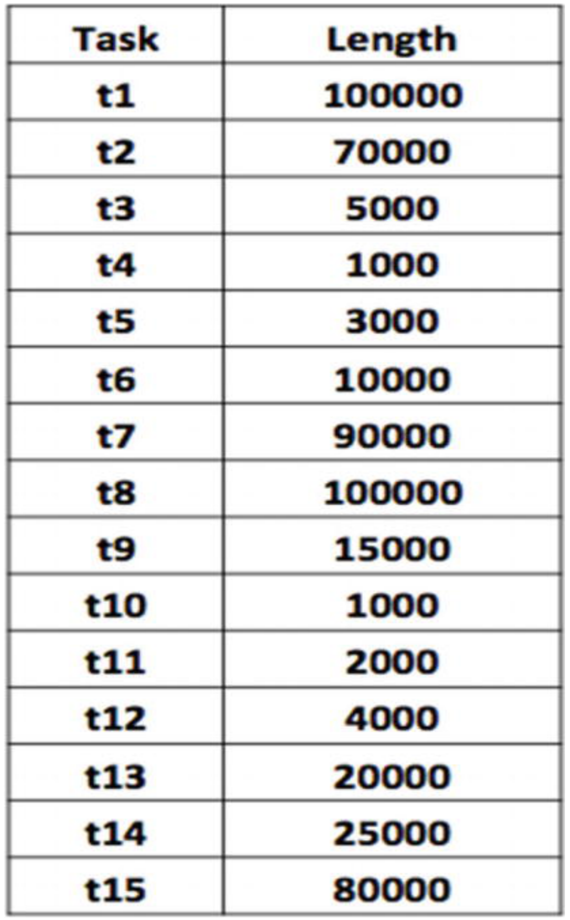
- Tasks have high waiting time.
- Not give any priority to tasks. That means when we have large tasks in the begin tasks list, all tasks must wait a long time until the large tasks to finish.
- Resources are not consumed in an optimal manner.
- In order to measure the performance achieved by this method,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

When applying FCFS, work mechanism will be as following



- Dot arrows refer to the 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.

Example: FCFS



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

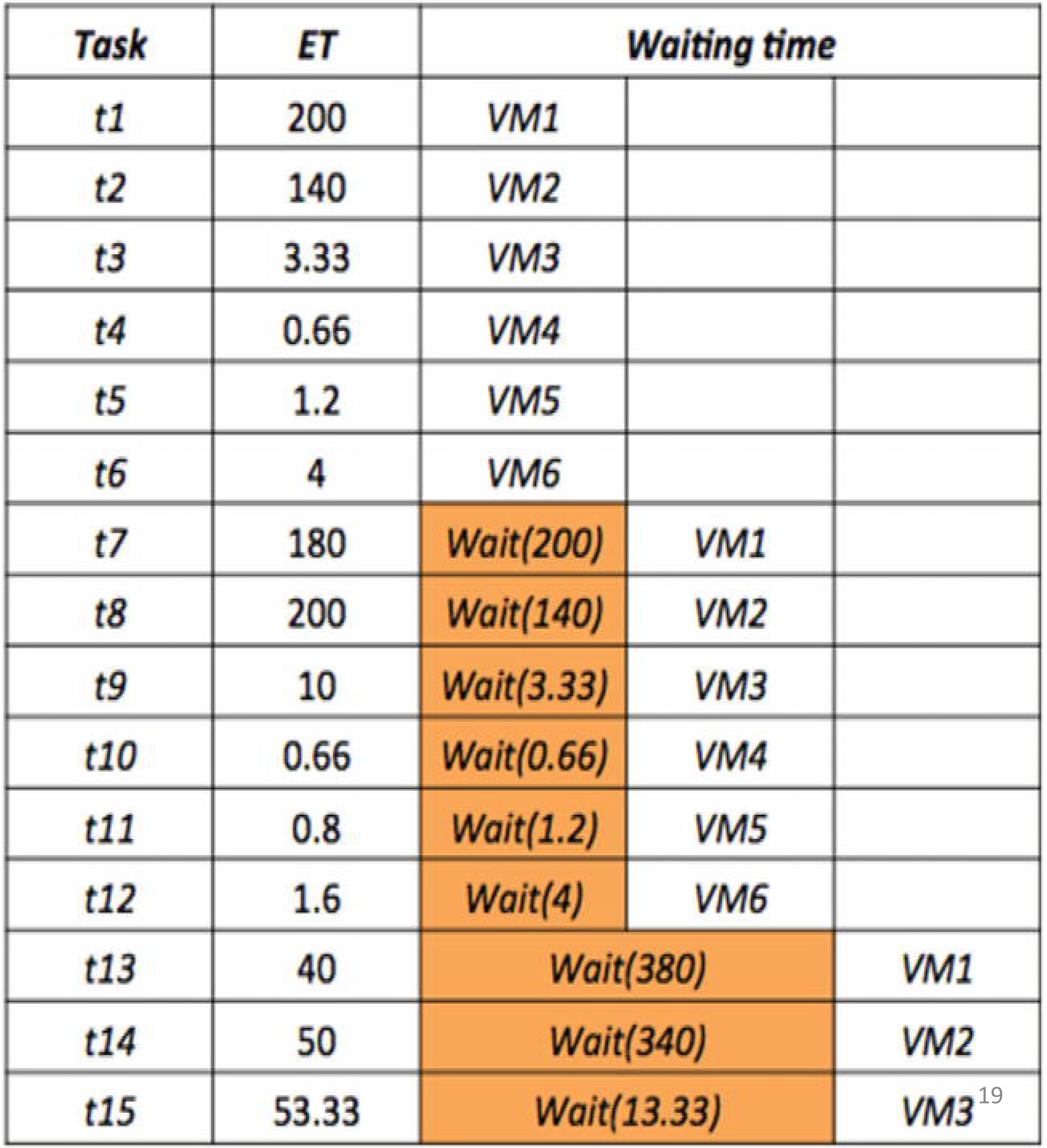
Assume we have six VMs with different properties based on tasks size:

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

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

Execution

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



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

SJF

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

Advantages

Wait time is lower than FCFS.

SJF has a minimum average waiting time among all task scheduling algorithms.

Disadvantages

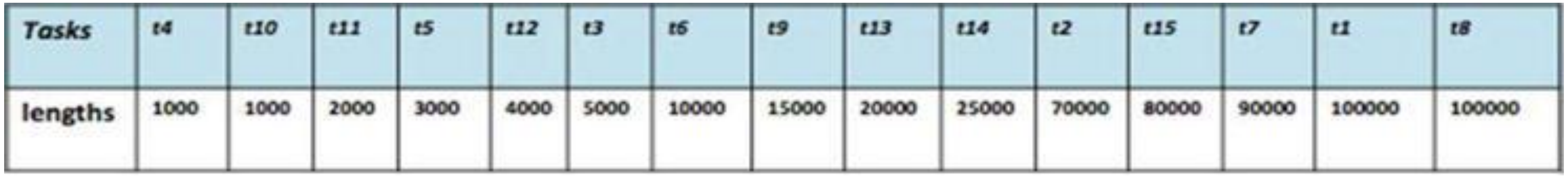
Unfairness to some tasks when tasks are assigned to VM, due to the long tasks tending to be left waiting in the task list while small tasks are assigned to VM.

Taking long execution time and TFT.

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, and then assigned to the VMs list sequentially.



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

Execution: SJF

VM1={t4⟶t6⟶t7}.

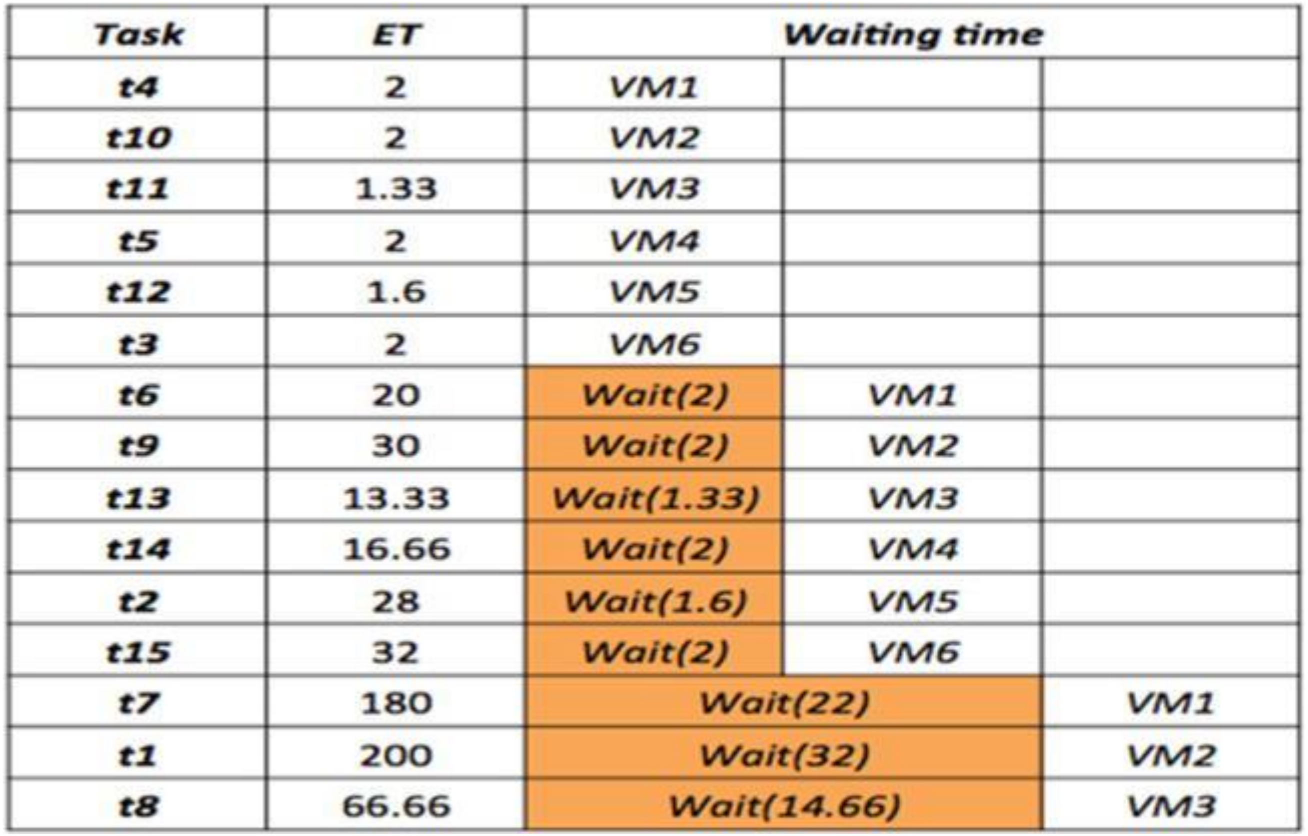
VM2={t10⟶t9⟶t1}.

VM3={t11⟶t13⟶t8}.

VM4={t5⟶t14}.

VM5={t12⟶t2}.

VM6={t3⟶t15}.



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

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 the VMs list.

Advantages

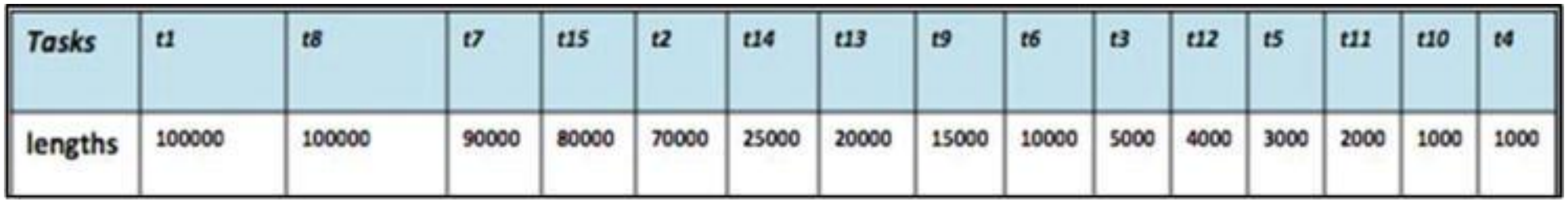
- Working to exploit the available resources in an efficient manner.
- This algorithm has better performance than the FCFS, SJF, and MIN-MIN algorithms

Disadvantages

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

Max-Min

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



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

Execution: Max-Min

VM6={t1⟶t13⟶t11}.

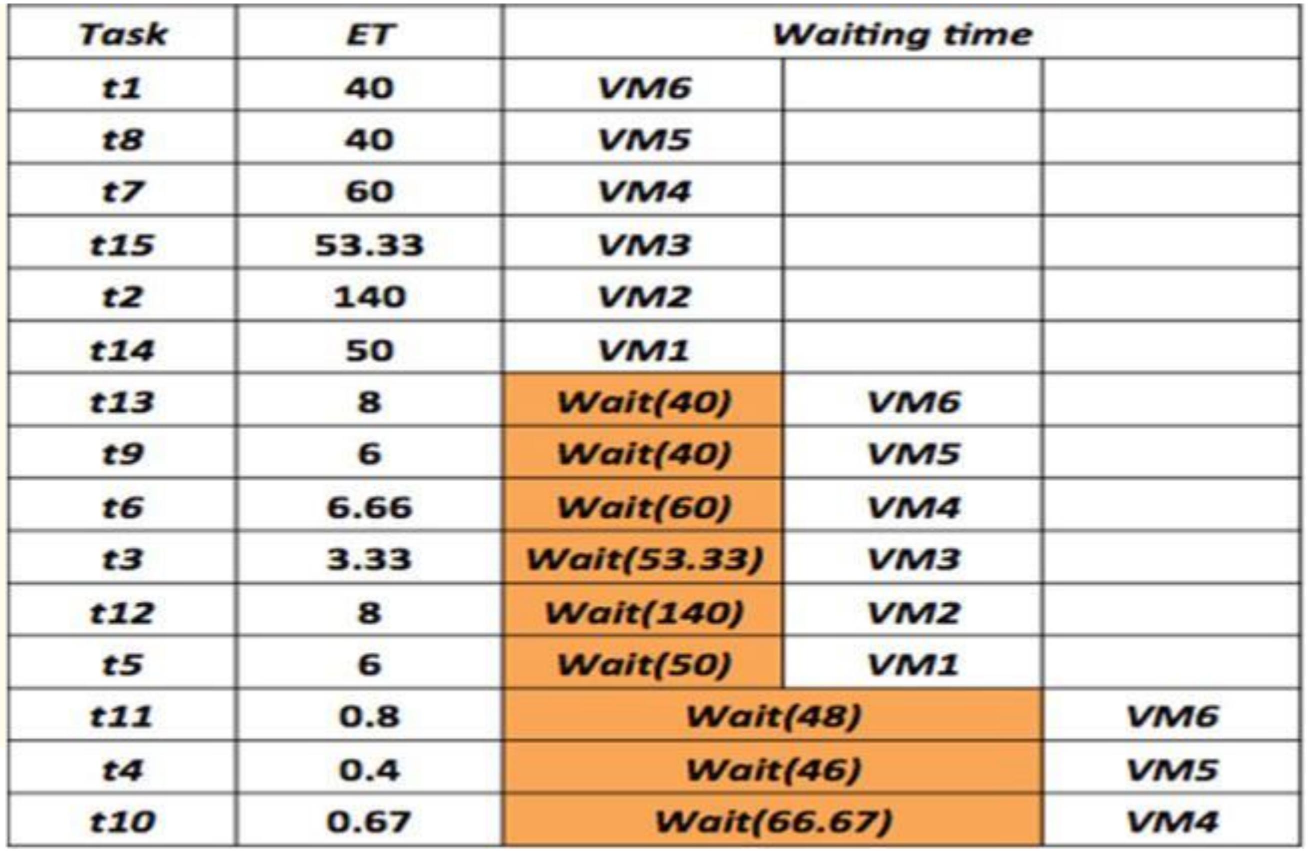
VM5={t8⟶t9⟶t10}.

VM4={t7⟶t6⟶t4}.

VM3={t15⟶t3}.

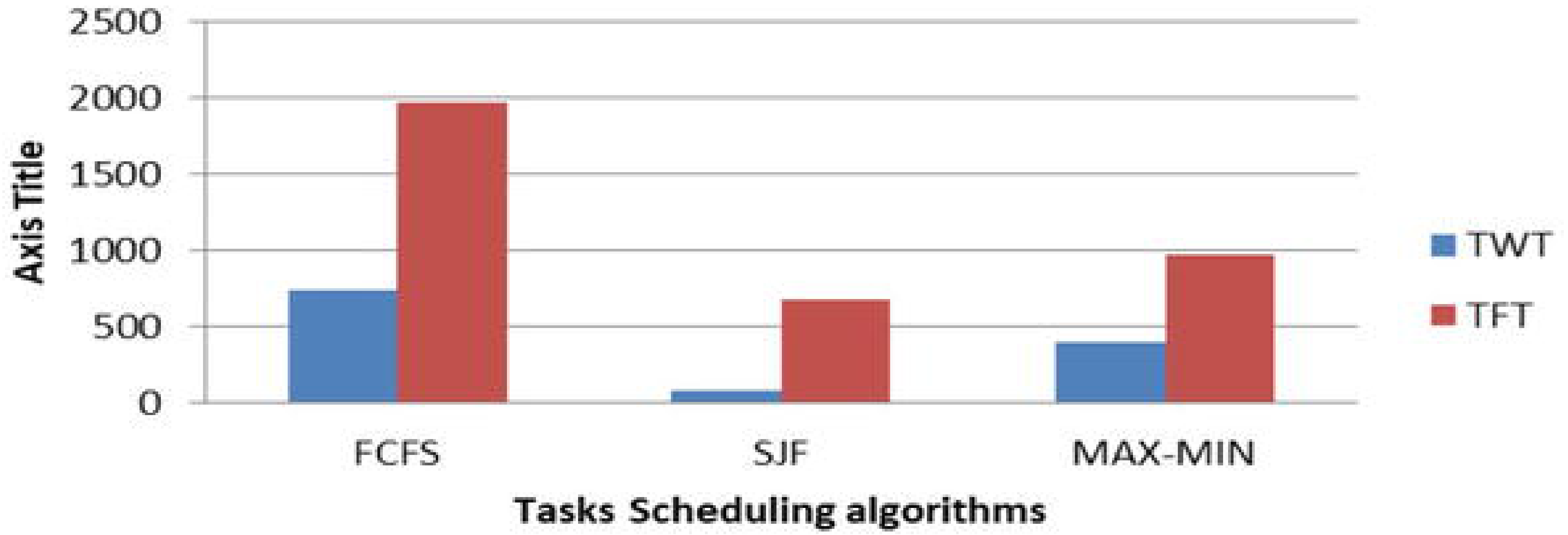
VM2={t2⟶t12}.

VM1={t14⟶t5}.



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

Result Comparison



Thank You