# DEMONSTRATING A SCHEDULING ALGORITHM USING CLOUD COMPUTING

NIDHI S. M.

Access Online

Article DOI:
10.5281/zenodo.21375004

## ABSTRACT

Cloud computing helps in sharing resources and running tasks easily using virtualization. Task scheduling is important to improve performance and save time. This project uses Python to implement two scheduling methods: First Come First Serve (FCFS) and Shortest Job First (SJF). The program takes task times as input and assigns them to two virtual machines. FCFS runs tasks in the same order they are entered, while SJF runs smaller tasks first. The output shows how tasks are divided and the total time taken by each virtual machine.

**KEYWORDS:** Cloud Computing, Scheduling Algorithm, FCFS, SJF, Virtual Machine, Task Scheduling, Python

## INTRODUCTION

Cloud computing is a technology that gives access to storage, servers, and applications through the internet. In cloud computing, many tasks can run at the same time using virtual machines. Good task scheduling is needed to make the system work faster and better. Scheduling algorithms decide how tasks are given to resources. Two common scheduling methods are First Come First Serve (FCFS) and Shortest Job First (SJF). FCFS runs tasks in the same order they are entered, while SJF runs the smaller tasks first. This paper shows a simple cloud scheduling system using Python. The system divides tasks between two virtual machines and shows how FCFS and SJF scheduling work.

### Objective

*A. To learn task scheduling in cloud computing*
This project helps to understand how tasks are handled in cloud computing. It shows how tasks are given and completed using virtual machines. Task scheduling helps the system work smoothly.

*B. To use FCFS and SJF scheduling in Python*
The project uses Python to create FCFS and SJF methods. FCFS runs tasks in the same order they are entered, while SJF runs smaller tasks first. This helps to see the difference between both methods.

*C. To divide tasks between virtual machines*
The program shares tasks between two virtual machines. This helps to reduce extra load on one machine. Sharing tasks properly can save time.

*D. To check execution time and task sharing*
The project shows how much time each virtual machine takes to finish tasks. It also shows how tasks are shared between them. This helps to compare the results of FCFS and SJF.

## PROPOSED METHODOLOGY

*A. User Input:* The user enters the time needed for four different tasks. These task times are taken as input by the program.

*B. Choosing the Scheduling Method:* The user selects either FCFS or SJF scheduling. In FCFS, tasks run in the same order as entered. In SJF, smaller tasks run first.

*C. Task Assignment:* The tasks are shared

Department of Electronics and Communication Engineering, Sharanbasava University, Kalaburagi, India.

How to Cite:
NIDHI S. M. (2026), Demonstrating a Scheduling Algorithm Using Cloud Computing, International Educational Journal of Science & Engineering (IEJSE),Vol: 9, Special Issue, May 2026 119-121

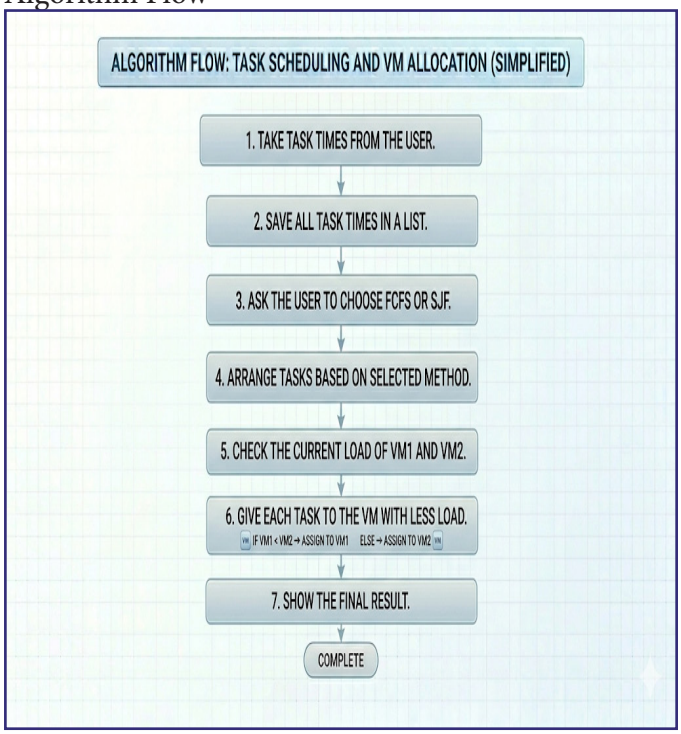
Research Paper

between two virtual machines: VM1 and VM2. Each new task is given to the virtual machine that has less total time so the work is shared properly.

*D. Showing the Output:* The program displays the order in which tasks are executed. It also shows the tasks given to VM1 and VM2 and the total time taken by each virtual machine.

## SYSTEM DESIGN

Algorithm Flow




**python implementation**

```
# step_1: taking user input for task duration
Task_1=int (input (“enter duration for Task_1:”))
Task_2=int (input (“enter duration for Task_2:”))
Task_3=int (input (“enter duration for Task_3:”))
Task_4=int (input (“enter duration for Task_4:”))

Tasks= [Task_1, Task_2, Task_3, Task_4]

#step_2: choosing algorithm
Choice=input (“FCFS or SJF? :”)

if Choice== “FCFS”:
   print (Tasks)

elif Choice== “SJF”:
   Tasks. sort()

else:
   print (“invalid choice”)
   exit ()

#step_3: scheduling tasks based on duration to virtual machines
VM1_time=0
VM2_time=0

VM1= []
VM2= []

for i in range (len (Tasks)):

   if VM1_time<=VM2_time:
      VM1.append(Tasks[i])
      VM1_time+=Tasks[i]

   else:
      VM2.append(Tasks[i])
      VM2_time+=Tasks[i]

#printing outputs
print(“----------OUTPUT----------”)
print (“code to be executed:”, Tasks)
print (VM1,”=”, VM1_time)
print (VM2,”=”, VM2_time)
```

VI. Sample output

Case 1: FCFS Scheduling

Input:

Task_1 = 5
Task_2 = 2
Task_3 = 8
Task_4 = 3
Choice = FCFS

Output:

Order of execution: [5, 2, 8, 3]
VM1 = [5, 3] = 8
VM2 = [2, 8] = 10

Case 2: SJF Scheduling

Input:

Task_1 = 5
Task_2 = 2
Task_3 = 8
Task_4 = 3
Choice = SJF

Output:

Order of execution: [2, 3, 5, 8]
VM1 = [2, 5] = 7

VM2 = [3, 8] = 11

**Advantages.**

A. *Simple and easy implementation:* The system is very easy to make using basic coding. It helps students understand the idea without difficulty.
B. *Shows basic cloud scheduling ideas:* It explains how tasks are managed in cloud computing. It is easy to see how work is given to virtual machines.
C. *Good sharing of work between virtual machines:* Tasks are divided between two virtual machines. This helps to keep the load balanced.
D. *Helps to understand FCFS and SJF:* It shows how FCFS and SJF work in a simple way. Students can easily see the difference between both methods.

**Disadvantages**

A. It works only for a small number of tasks, so it is not good for large real cloud systems. It cannot handle heavy or big workloads.
B. It uses only FCFS and SJF methods, so it is limited. It does not include other advanced scheduling techniques used in real systems.

**Applications**

A. *Cloud resource management:* It helps in sharing and managing cloud resources in a simple way. It shows how tasks are given to different systems.
B. *Data centers:* It helps in understanding how work is handled in data centers. It shows how tasks are processed and managed.
C. *Distributed computing systems:* It shows how tasks are divided across multiple systems. This helps to complete work in a balanced way.
D. *Task execution monitoring:* It helps to track how tasks are running. It shows the order and time taken for each task.
E. *Educational demonstrations of scheduling algorithms:* It helps in learning how scheduling works. It clearly shows FCFS and SJF in a simple way.

## CONCLUSION

This paper presents a simple demonstration of scheduling algorithms in cloud computing using Python. The system implements FCFS and SJF algorithms and distributes tasks between two virtual machines. The project helps to understand the importance of scheduling in cloud systems. It also shows how workload balancing improves resource usage and overall performance. In the future, more scheduling methods such as Round Robin and Priority Scheduling can be added to improve the system.

## REFERENCES

1. Rajkumar Buyya, "Cloud Computing Principles and Paradigms," Wiley Publications, 2011.
2. Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, "Operating System Concepts," 9th Edition, Wiley, 2013.
3. Kai Hwang, Geoffrey Fox, and Jack Dongarra, "Distributed and Cloud Computing," Morgan Kaufmann, 2012.
4. Python Software Foundation, "Python Documentation," Available: https://www.python.org/
5. A. Tanenbaum, "Modern Operating Systems," Pearson Education,2014