Task Scheduling Algorithm in Cloud Computing

Dhaval Dadhaniya
18it025@charusat.edu.in

Prof. Nehal Patel
nehalpatel.it@charusat.ac.in

Abstract-

In today's world Cloud computing is one of the most demanding and most emerging technology. In cloud computing the user has to pay for what they have used for what period of time i.e., no upfront cost for the resources. Most of the companies are moving towards cloud computing instead of maintaining physical servers and data centres because of its high potential and flexibility and profitability. But due to huge volume of requests of resources it is difficult to manage resource optimally. Therefore, scheduling is necessary to optimally deliver resources to users. In this paper we are going to talk about some of the Task Scheduling Algorithms in Cloud Computing.

Keywords- Cloud Computing, Algorithms, Task Scheduling, Fog Node, Edge Node, Cloud Node.

I. Introduction

Recently Information Technology and Internet are ruling the world with its demand in each and every aspect of life. Latest technology developed in the field of IT is Cloud Computing. Most of the big companies are taking Cloud Computing into consideration because it is providing on-demand flexible infrastructure and profitable computing services. In Cloud Computing there are 3 types of services available: IaaS, PaaS and SaaS. These all services are available for the users in pay-as-you go model. A sample architecture is shown in Fig. 1.

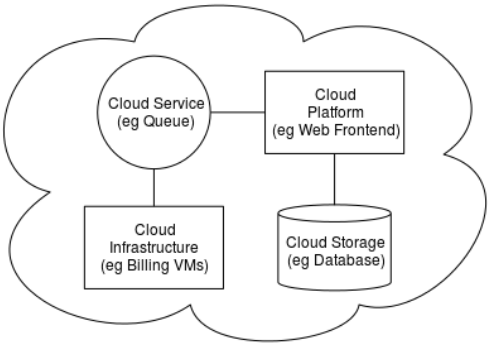

Cloud Service (eg Queue)
Cloud Platform (eg Web Frontend)
Cloud Infrastructure (eg Billing VMs)
Cloud Storage (eg Database)

Fig. 1 Sample Cloud Architecture

Due to increase in use of Cloud Computing services it is very difficult to provide resources optimally, so to provide these resources in an optimal way we use Task Scheduling Algorithms.

Task Scheduling Algorithms is the assignment of start and end times to a set of tasks, subject to certain constraints and to maximize the CPU utilization. There are many advantages of Task Scheduling Algorithms.

1. It is easy to manage cloud computing performance and Quality of Service.
2. We can achieve high system throughput.
3. We can improve Load Balancing.
4. If the scheduling algorithm is optimized then we can

maximize the resource use and can decrease execution time.

5. We can manage CPU and memory load using Scheduling Algorithms.

II. Task Scheduling Algorithms Categories

Task Scheduling Algorithms are divided into many categories which can be seen in below figure. 2.

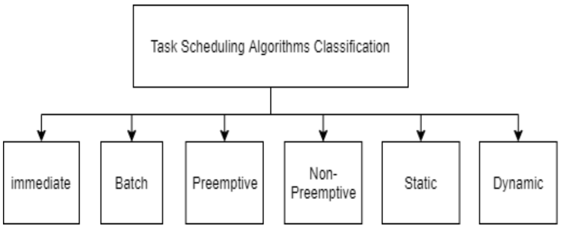


Fig. 2 Categories of Algorithms

Immediate Scheduling: It schedules task immediately like if task arrives it is scheduled to the VM.

Batch Scheduling: In these tasks are grouped and then that particular group is scheduled to VM.

Pre-emptive Scheduling: In this task can be interrupted and can be scheduled to another VM.

Non-Pre-emptive: In this another task waits until the first task finishes its execution.

In this paper the focus will be on 3 algorithms 1.Best Fit 2.First Fit 3.Round Robin.

But Before that let’s see how task scheduling works in Cloud Computing by following figure.

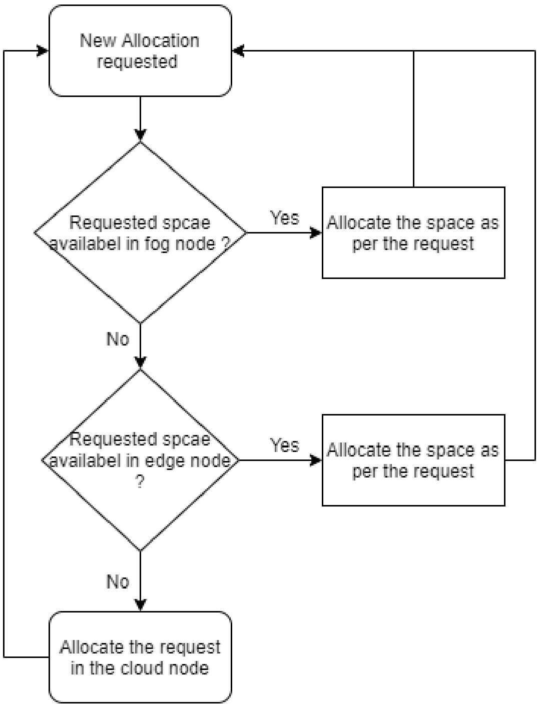

New Allocation requested
Requested spcae availabel in fog node ?
Yes
Allocate the space as per the request
No
Requested spcae availabel in edge node ?
Yes
Allocate the space as per the request
No
Allocate the request in the cloud node

Fig. 3 Flow of Task

In the above figure we can see that a request arrives and first of all it checks for the fog-node if there is some empty space then it allocated that particular task to fog-node resource. Then if there is no space in fog-node then it goes to edge-node resource. If there is not enough memory to execute that particular task then it goes to the cloud node and finally there it gets executed. In this paper discussion of three Scheduling algorithms is done.

A. Best Fit Algorithm
B. First Fit Algorithm
C. Round Robin Algorithm.

A. Best Fit Scheduling Algorithm

Best fit algorithm is memory management algorithm. In Best Fit algorithm the tasks are kept in the smallest memory block which is available. e.g. If first request is of 500K and the first block is of 1000K and second block is of 600K then the task will be scheduled to 600K block. In this case first of all the task will find space in the fog-node, is space available in fog - node? If yes then the task will allocate to the resource available in the fog-node.

The working of the Best-Fit algorithm is mentioned below:

Step 1: Load the data set file into the program in which the number of process and the burst time of the task is mentioned.

Step 2: Run the program and the scheduled task can be seen as an output.

The working of best-fit algorithm is given in the figure below.

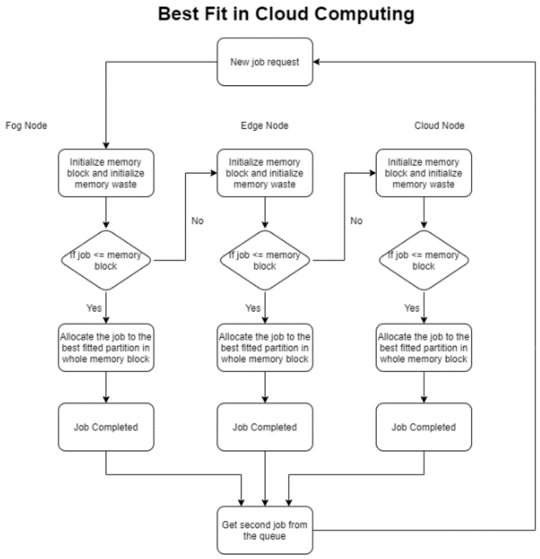

Best Fit in Cloud Computing
New job request
Fog Node
Edge Node
Cloud Node
Initialize memory block and initialize memory waste
Initialize memory block and initialize memory waste
Initialize memory block and initialize memory waste
No
No
If job <= memory block
If job <= memory block
If job <= memory block
Yes
Yes
Yes
Allocate the job to the best fitted partition in whole memory block
Allocate the job to the best fitted partition in whole memory block
Allocate the job to the best fitted partition in whole memory block
Job Completed
Job Completed
Job Completed
Get second job from the queue

Fig. 4 Best-fit Algorithm

Now let’s see the implementation if the best-fit algorithm.

Step - 1 New Job request arrives.

Step - 2 Memory block and the memory waste block gets initialize.

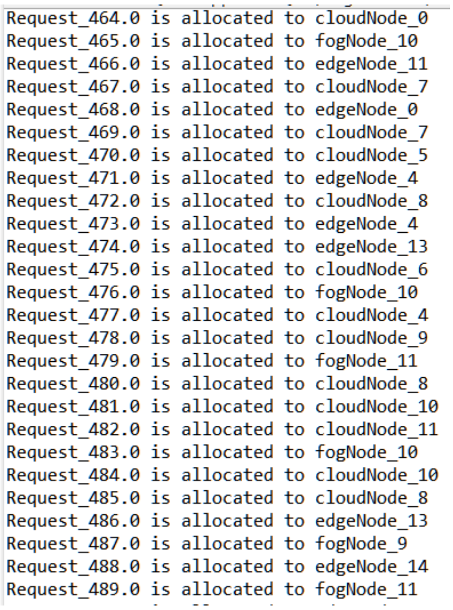
Step - 3 It checks for the memory block in the fog-node if space is available then it is the task is scheduled to the resources available in the fog-node, if not then the task goes to edge-node if the space is available there then task is scheduled in edge-node and at last if space is not available in the edge-node also then the task is scheduled in the cloud-node.

Step - 4 if the task is scheduled in the cloud node then the steps are repeated with another task in the requested queue.

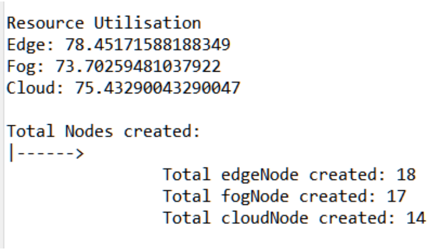
Advantage of best-fit algorithm is it is memory efficient because it allocates task to the minimum possible block.

Disadvantage of best-fit algorithm is it is slow, so it takes more time to execute as it traverse through all blocks to find optimal block for allocation.

Here in the figure shown below is the output of the Best-Fit Algorithm.



```
Request_464.0 is allocated to cloudNode_0
Request_465.0 is allocated to fogNode_10
Request_466.0 is allocated to edgeNode_11
Request_467.0 is allocated to cloudNode_7
Request_468.0 is allocated to edgeNode_0
Request_469.0 is allocated to cloudNode_7
Request_470.0 is allocated to cloudNode_5
Request_471.0 is allocated to edgeNode_4
Request_472.0 is allocated to cloudNode_8
Request_473.0 is allocated to edgeNode_4
Request_474.0 is allocated to edgeNode_13
Request_475.0 is allocated to cloudNode_6
Request_476.0 is allocated to fogNode_10
Request_477.0 is allocated to cloudNode_4
Request_478.0 is allocated to cloudNode_9
Request_479.0 is allocated to fogNode_11
Request_480.0 is allocated to cloudNode_8
Request_481.0 is allocated to cloudNode_10
Request_482.0 is allocated to cloudNode_11
Request_483.0 is allocated to fogNode_10
Request_484.0 is allocated to cloudNode_10
Request_485.0 is allocated to cloudNode_8
Request_486.0 is allocated to edgeNode_13
Request_487.0 is allocated to fogNode_9
Request_488.0 is allocated to edgeNode_14
Request_489.0 is allocated to fogNode_11
```



```
Resource Utilisation
Edge: 78.45171588188349
Fog: 73.70259481037922
Cloud: 75.43290043290047

Total Nodes created:
|------>
                Total edgeNode created: 18
                Total fogNode created: 17
                Total cloudNode created: 14
```

B. First-Fit Scheduling Algorithm

This is the simplest algorithm for scheduling tasks in. In this algorithm task is allocated in the basis of first come first serve. e.g. if task t1 arrives and having 600K of memory and task t2 having 100K and the first two blocks are of 800K and 1000K then task t1 is assigned at first block and task t2 is assigned to second block.

It is mostly like best-fit algorithm but the only difference is in best-fit according to algorithm it uses the smallest block possible to allocate to the task and in first-fit, the task is allocated on first come basis.

Here is the flow of the First-Fit Algorithm in the figure down below.

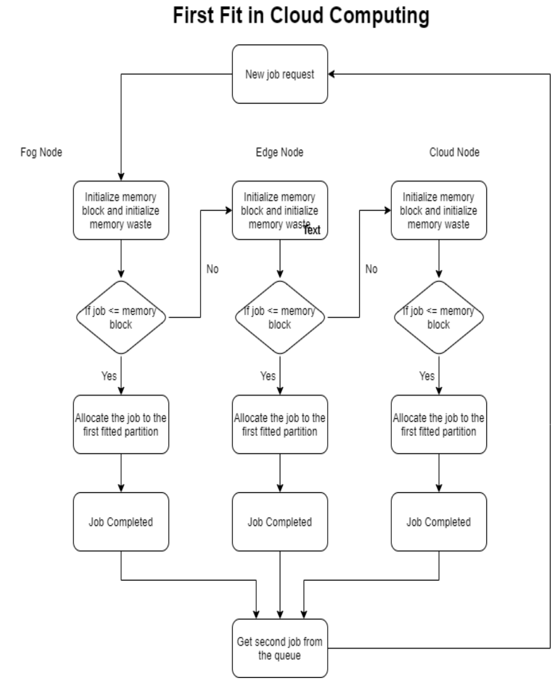
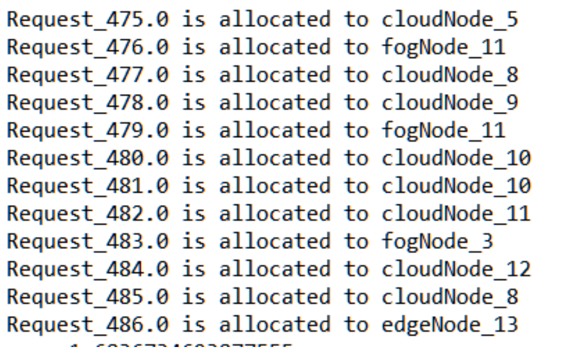
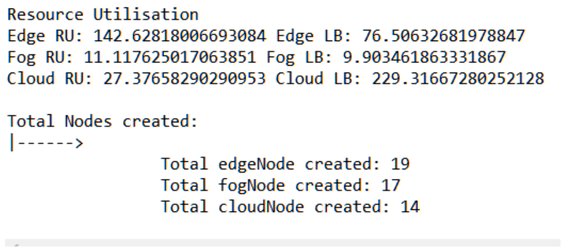

First Fit in Cloud Computing
New job request
Fog Node
Edge Node
Cloud Node
Initialize memory block and initialize memory waste
Initialize memory block and initialize memory waste
Text
Initialize memory block and initialize memory waste
No
No
If job <= memory block
If job <= memory block
If job <= memory block
Yes
Yes
Yes
Allocate the job to the first fitted partition
Allocate the job to the first fitted partition
Allocate the job to the first fitted partition
Job Completed
Job Completed
Job Completed
Get second job from the queue

Fig. 6 First-Fit Scheduling Algorithm

Here is the output of the all the tasks scheduled using First-Fit Scheduling Algorithm.



```
Request_475.0 is allocated to cloudNode_5
Request_476.0 is allocated to fogNode_11
Request_477.0 is allocated to cloudNode_8
Request_478.0 is allocated to cloudNode_9
Request_479.0 is allocated to fogNode_11
Request_480.0 is allocated to cloudNode_10
Request_481.0 is allocated to cloudNode_10
Request_482.0 is allocated to cloudNode_11
Request_483.0 is allocated to fogNode_3
Request_484.0 is allocated to cloudNode_12
Request_485.0 is allocated to cloudNode_8
Request_486.0 is allocated to edgeNode_13
```



```
Resource Utilisation
Edge RU: 142.62818006693084 Edge LB: 76.50632681978847
Fog RU: 11.117625017063851 Fog LB: 9.903461863331867
Cloud RU: 27.37658290290953 Cloud LB: 229.31667280252128

Total Nodes created:
|------>
                Total edgeNode created: 19
                Total fogNode created: 17
                Total cloudNode created: 14
```

Advantage of first-fit algorithm is, it is faster then best-fit algorithm as it allocates the task to the nearest possible block available.

Disadvantage of first-fit algorithm is, memory waste. It can allocate large memory block to small process if small process is ahead of large process.

C. Round-Robin Task Scheduling Algorithm

The name Round-Robin suggest that it is comes from the Round-Robin principal. This algorithm is used for multitasking.

In Round-Robin each task from the request queue is executed for a certain interval of time known as time quantum in cyclic manner.

Here is the flow of Round-Robin in the figure given below.

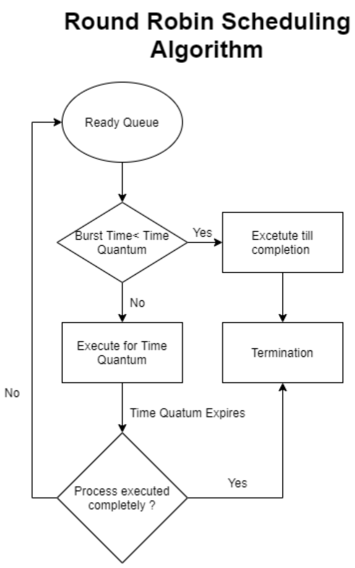


Fig. 8 Round-Robin Scheduling Algorithm

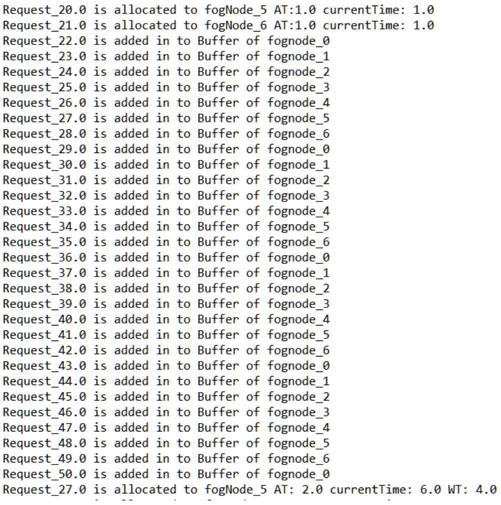
Working of Round-Robin Algorithm is as follows:

Step-1 Tasks arrives in ready queue.

Step-2 System checks if the burst time is smaller than time quantum if yes then then checks for the memory in fog-node, edge-node and cloud node and completes its execution.

Step-3 If the burst time is more than the time quantum then the task is executed for that time quantum and then next task arrives and executes in cyclic manner.

Figure down below is the tasks executed using Round-Robin Scheduling Algorithm.



```
Request_20.0 is allocated to fogNode_5 AT:1.0 currentTime: 1.0
Request_21.0 is allocated to fogNode_6 AT:1.0 currentTime: 1.0
Request_22.0 is added in to Buffer of fognode_0
Request_23.0 is added in to Buffer of fognode_1
Request_24.0 is added in to Buffer of fognode_2
Request_25.0 is added in to Buffer of fognode_3
Request_26.0 is added in to Buffer of fognode_4
Request_27.0 is added in to Buffer of fognode_5
Request_28.0 is added in to Buffer of fognode_6
Request_29.0 is added in to Buffer of fognode_0
Request_30.0 is added in to Buffer of fognode_1
Request_31.0 is added in to Buffer of fognode_2
Request_32.0 is added in to Buffer of fognode_3
Request_33.0 is added in to Buffer of fognode_4
Request_34.0 is added in to Buffer of fognode_5
Request_35.0 is added in to Buffer of fognode_6
Request_36.0 is added in to Buffer of fognode_0
Request_37.0 is added in to Buffer of fognode_1
Request_38.0 is added in to Buffer of fognode_2
Request_39.0 is added in to Buffer of fognode_3
Request_40.0 is added in to Buffer of fognode_4
Request_41.0 is added in to Buffer of fognode_5
Request_42.0 is added in to Buffer of fognode_6
Request_43.0 is added in to Buffer of fognode_0
Request_44.0 is added in to Buffer of fognode_1
Request_45.0 is added in to Buffer of fognode_2
Request_46.0 is added in to Buffer of fognode_3
Request_47.0 is added in to Buffer of fognode_4
Request_48.0 is added in to Buffer of fognode_5
Request_49.0 is added in to Buffer of fognode_6
Request_50.0 is added in to Buffer of fognode_0
Request_27.0 is allocated to fogNode_5 AT: 2.0 currentTime: 6.0 WT: 4.0
```

Advantage if Round-Robin algorithm is each and every process gets equal part of the resource.

Disadvantage of Round-Robin algorithm is the waiting time and the response time is very high and the throughput is low.

Reference:

https://www.intechopen.com/books/scheduling-problems-new-applications-and-trends/types-of-task-scheduling-algorithms-in-cloud-computing-environment

https://sci-hub.se/10.1109/icabcd.2019.8851045