A Study on Job Scheduling Algorithms in Cloud Computing

P. Nandhini

Abstract—Cloud computing is a system of conveying technology to the user by using servers of Internet for processing and storage of data, while the client system uses the data. The great costs of development in customer requirements, network platforms, data pressures and volumes on response time pushed companies to transfer to Cloud Computing providing on demand internet presented services of IT. Job scheduling is a most significant part in the environment of cloud. With the growing number of users, Job scheduling grows into an active task. Ordering the jobs by scheduler while keeping the stability between qualities of services (QoS), fairness and efficiency of jobs are quite interesting. Scheduling algorithms are executed considering parameters such as resource utilization, priority, computational time, bandwidth, resource availability, throughput, latency, cost, physical distances and performance. This paper discussed about the scheduling algorithms and policies of cloud computing.

Keywords—Cloud Computing, Scheduling Process, Job Scheduling, Job Scheduling Algorithms

I. INTRODUCTION

Cloud computing is a kind of computing that depend on pooled computing resources quite than taking personal devices or local servers to handle the applications. Storage, applications and other services are accessed through the Web. The services are distributed and used over the Internet and are paid for by the customer of cloud on pay-per-use business model. Consumers and organizations have various reasons for selecting to use the services of cloud computing. They might include Security, high availability, scalability, convenience, simplicity, low costs and anytime, anywhere access. In the past, software had to be sent on a CD-ROM, and updates had to be downloaded later to keep the software bug-free and secure, cloud computing lets vendors to distribute software and services over the Internet without the necessity for traditional media or installation. An interrelated idea is called Software as a Service, or SAAS. When the Internet, as a network, became skillful of conveying large amounts of data in a small period of time, it became likely to offer not only simple web pages but also entire services online. The system of client could be a laptop computer, a desktop computer, a TV, or a phone, Computing power. Thin clients can be either hardware or software based.

As an enormous number of clients share the resources of cloud and dispatch their tasks to the cloud, it has turned out to be a challenge to schedule these tasks. Therefore, job scheduling is a sizzling topic in distributed and cloud computing.

As cloud technology is growing day by day and meets numerous challenges, one of them is open is scheduling. Scheduling is fundamentally a set of constructs made to have a supervisory hand over the order of work to be completed by a computer system. Algorithms are dynamic to schedule the jobs for the process of execution. Job scheduling algorithm is one of the greatest challenging hypothetical problems in the area of cloud computing. Several deep inquiries have been carried out in the domain area of job scheduling of cloud computing. This survey aims at study about job scheduling algorithms under cloud environment, namely, First Come First Serve (FCFS), Round Robin (RR), Shortest Job First Scheduling (SJF), Priority Scheduling, Min-Min, Max-Min, Genetic algorithms in terms of their ability to provide quality service for the task and guarantee. This paper also includes the comparative study on above job scheduling algorithms.

II. SCHEDULING PROCESS

The major goal of scheduling is to maximize the utilization of resource and reduce processing time of the tasks. The scheduler should order the jobs in a way where stability among is improving the quality of services and simultaneously sustaining the fairness and efficiency between the jobs. In the cloud, the scheduling process can be done in three stages:

a. Resource discovering and filtering – Datacenter Broker be familiar with the recent status of all the resources that are existing in the cloud and also the left over resources that may be available. These resources are usually the Virtual machines. It often assembles the status of each resource involved in the cloud.

b. Resource selection – Based on information attained from the resource status concerning current queued jobs and information on the status of resources of cloud, the scheduler of cloud makes decisions regarding the deletion or creation of specific cloud nodes (VMs) to best suit the set of jobs waiting to be run.
c. Job submission – Finally the job is submitted to the selected available resource.

III. JOB SCHEDULING IN CLOUD

Job Scheduling of cloud computing refers to sending off the computing tasks to resource pooling among dissimilar resource users according to definite rules of resource use below a specified circumstances of cloud. In cloud computing, presently there is not a uniform standard for job scheduling. Job scheduling and Resource management are the key technologies of cloud computing that plays a very important role in a proficient resource management of cloud.

The Job management is the basic concept of cloud computing systems; task scheduling problems are most important which relates to the effectiveness of the entire cloud computing system. Job scheduling is a mapping mechanism from users’ tasks to the correct selection of resources and its implementation. Job scheduling is convenient and flexible. Jobs and job streams can be scheduled to run every time required, based on needs, business functions, and priorities [4]. Job streams and processes can set up daily, weekly, monthly, and yearly in advance, and run on requiring jobs with no need for help from support staff.

IV. JOB SCHEDULING ALGORITHMS

4.1 First Come First Serve Scheduling Algorithm:

It is well-known as First in First out. Shortest job next is useful for its easiness and because it decreases the average time amount, each process has to wait till its execution is finished. It is one of the different Scheduling algorithms we have at allocate the CPU in the order in which the process reach. It’s expected that organized queue is managed such as first in first out which means that the job which arrive first will be treated first lacking other preferences [5].

Algorithm FCFS:

Step 1: Initialize the Job.

Step 2: First job allotted to the queue and increase jobs up to n numbers.

Step 3: Add next job ‘I’ at last location of the main queue.

Figure 1 shows the process of the FCFS algorithm with an example.

<table>
<thead>
<tr>
<th>Process</th>
<th>Arrival Time</th>
<th>Execution Time</th>
<th>Service Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>P2</td>
<td>2</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>

4.2 Shortest Job First Scheduling Algorithm:

Shortest job First (SJF) or Shortest Job Next (SJN) or Shortest Process Next (SPN) is a scheduling technique that chooses the job with the minimum execution time. The jobs are queued with the minimum execution time engaged first and the job with the longest execution time engaged last and given the last priority. This algorithm is contracted with different approach in this algorithm CPU is assigned to the process with minimum burst time. Figure 2 shows the process of Non-Preemptive SJF algorithm with an example and Figure 3 shows the process of Preemptive SJF algorithm with an example.

Figure 2. Example of Non-Preemptive SJF

Figure 3. Example of the Preemptive SJF
Algorithm SJF:
for i = 0 to i < main queue-size
  if job i+1 length < job i length then
    add job i+1 in front of job i in the queue
  end if
  if main queue-size = 0 then
    job i last in the main queue
  end if
end for

4.3 Round-Robin Scheduling Algorithm:
It is one of the reasonable, simplest, ancient and most broadly used scheduling algorithms, designed particularly for the system of timesharing. A small unit of time, named quantum or time slices is defined. All runnable processes are saved in a circular queue. The scheduler of the CPU goes around this queue, assigning the CPU to every process for a time intermission of one quantum. Additional processes are added to the end of the queue \[5\]. The scheduler of CPU selects the first process from the queue, sets a clock to intrude after one quantum, and the process is dispatched. The CPU is prevented if the process is still running at the completion of the quantum, and the process is added to the end of the queue. If the process completed before the end of the quantum, the process itself voluntarily discharges the CPU. Figure 4 shows the process of the Round Robin algorithm with an example.

Algorithm RRS:
Step 1: Have the arranged queue as a FIFO queue of processes.
Step 2: Fresh processes are added to the end of the organized queue.
Step 3: The scheduler of CPU picks the first process from the organized queue, sets a clock to disturb after 1 time slot, and dispatches the process.
Step 4: The process may need a CPU burst of a smaller amount than 1 time quantum.
  a. In this instance, the process itself will discharge the CPU voluntarily.
  b. The scheduler will then continue with the following process in the ready queue.
Step 5: Or else, if the CPU burst of the presently running process is lengthier than 1 time quantum,
  a. The timer will explode and will cause a disturb to the OS
  b. A context switch will be implemented, and the process will be put at the end of the ready queue.
c. The scheduler of CPU will then choose the following process in the ready queue.

4.4 Priority Scheduling Algorithm:
This Scheduling algorithm is proactive in which everything is based on the precedence in this scheduling algorithm every process of the system is based on the precedence whereas maximum priority job can run first, whereas lesser priority job can be made to wait, the major problem of this algorithm is starvation of a process \[6\]. Figure 5 shows the process of Priority scheduling algorithm with an example.

Algorithm PSA:
for i = 0 to i < main queue-size
  if priority (job i+1) > (job i) then
    add job i+1 in front of job i in the queue
  end if
end for

4.5 Genetic Algorithm (GA):
It is a problem solving method that uses genetics as its model of problem solving. It is a search technique to discover an improved solution. GA handles a populace of probable solution. Every solution is denoted through a chromosome. A Genetic algorithm is a scheduling method in which the tasks are allotted resources according to separate solutions (which are named as schedules in context of scheduling), which says about which resource is to be allocated to which task.
It is based on the biological concept of populace generation. In GA the primary population is made randomly. It is a random searching method\(^7\). Figure 6 shows the process of Genetic algorithm as Flowchart.

Algorithm Genetic Algorithm:
- Initialization: Great initial random population
- Evaluate
- Keep the best
- While termination not true do
  - Selection
  - Crossover
  - Mutation
  - Evaluate
  - Elitist
- Check exit
- End while
- Return mapping result.

iii. Crossover Function – The crossover operators are the best significant component of any evolutionary-like algorithm. It generates the new offspring by moving to the improved state to parents. It also casually selects two parent chromosomes.

iv. Mutation Function – This function is designed to decrease the processor’s ideal time that is waiting for the data from new processors. There are numerous mutation operators like Swap, Move, Rebalancing and Move and Swap.

v. Evaluation Function – Evaluation depends on the time of execution and cost of execution. Those schedules will be nominated for next generation whose execution time and cost is less.

4.6 Max-Min and Min-Min algorithm:

In these algorithm’s first least execution time and smallest completion time for all the tasks are intended. MET (Minimum Execution Time): MET allot each task to that resource which executes tasks in the smallest amount of time. But it is not considered that resources were presented at that time or not. MCT (Minimum Completion Time): MCT allot all tasks to those resources which finish them in the smallest amount of completion time. It means this allocate some tasks to those resources that do not have less execution time\(^8\). Min-Min: Min-Min begins with the set of all unassigned tasks in the makespan. This algorithm work in two phases: First, the least predictable completion time for all the tasks is intended. The completion time for all the tasks is intended for all the machines. In the second phase, the task with the least expected completion time from makespan is chosen and that task allocated to the corresponding resource. Then the task which is finished that is detached from the makespan and this process is repetitive until all tasks are finished. Fig.3 shows fundamental Max-min algorithm, rj stand for the ready time of resource Rj to perform a task, Cij and Eij stand for the anticipated completion time and task’s execution time\(^9\). Then the task Tk to highest expected completion time is chosen. That task allocated to resource Rj that finishes a task at least execution time.

Algorithm

- **Step 1.** for all submitted tasks in meta-task; Ti
- **Step 2.** for all resources; Rj
- **Step 3.** Cij = Eij + rj
- **Step 4.** While meta-task is not empty
- **Step 5.** find the task Tk consumes maximum completion time.

![Figure 6. Flowchart of Genetic Algorithm](image-url)
National machines are freedom to schedule and this algorithm is the short jobs scheduled first, until the extensive jobs will go after the small jobs. The inadequacy of Min algorithm will perform short jobs in similar and the which can finish the task in the minimum possible time. Min-min can cause both the entire batch jobs executed time get unbalanced and longer load. Even extensive jobs cannot be executed. Compared with the traditional Min-min algorithm, enhanced algorithm adds the three constraints (the dynamic priority model, quality of service and the cost of service) strategy which can modify this condition. The experimental results of enhanced Min-Min algorithm prove it can raise the utilization rate of resource; extensive tasks can execute at reasonable times and meet the requirements of users.

Step 6. assign Tk to the resource Rj which gives minimum execution time

Step 7. remove Tk from meta-tasks set

The major idea of the Min-Min algorithm is as rapidly as possible to send off each task to virtual machines as resources which can finish the task in the minimum possible time. Min-Min algorithm will perform short jobs in similar and the extensive jobs will go after the small jobs. The inadequacy of this algorithm is the short jobs scheduled first, until the machines are freedom to schedule and perform extensive jobs.

TABLE 1 COMPARATIVE STUDY ON JOB SCHEDULING ALGORITHMS

<table>
<thead>
<tr>
<th>Scheduling Algorithms</th>
<th>Parameters</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Objectives</th>
<th>Complexity</th>
<th>Allocation</th>
<th>Type of system</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Come First Serve</td>
<td>Arrival Time</td>
<td>Easy to implement</td>
<td>Doesn’t consider any other criteria for scheduling</td>
<td>Allocation of CPU according to the task Arrival</td>
<td>Simplest Scheduling Algorithm</td>
<td>CPU is allocated in the order in which the processes arrive</td>
<td>Suitable for Batch system</td>
</tr>
<tr>
<td>Round Robin</td>
<td>Arrival time, Time quantum</td>
<td>Less complexity and load is balanced more fairly</td>
<td>Pre-emption is required</td>
<td>Preemption after fixed time</td>
<td>Performance depends upon the size of time quantum</td>
<td>The preemption takes place after a fixed interval of time</td>
<td>Suitable for time sharing system</td>
</tr>
<tr>
<td>Shortest Job First Scheduling</td>
<td>Arrival Time, Completion Time, Turn Around Time, Waiting Time</td>
<td>The throughput is increased because of more process executed in less amount of time</td>
<td>Longer processes have more waiting time, suffer by starvation</td>
<td>Effective Resource Allocation</td>
<td>Difficult to understand and code</td>
<td>CPU is allocated to the process with least CPU burst time</td>
<td>Suitable for Batch system</td>
</tr>
<tr>
<td>Priority Scheduling</td>
<td>Waiting time, response time, total completion time</td>
<td>The priority of a process selected based on memory requirement, time requirement or user preference.</td>
<td>A second scheduling algorithm is required to schedule the processes which have same priority</td>
<td>Priority Job allocation</td>
<td>Difficult to understand</td>
<td>Based on priority, So the higher priority job can run first</td>
<td>Suitable for both Batch and time sharing systems</td>
</tr>
<tr>
<td>Min-Min</td>
<td>Makespan, Expected completion time</td>
<td>Better makespan compared to other algorithms</td>
<td>Poor load balancing</td>
<td>To promise the guarantee regarding the provided resources.</td>
<td>Execution time reduced</td>
<td>Smaller tasks will execute first.</td>
<td>Distributed systems</td>
</tr>
<tr>
<td>Max-Min</td>
<td>Make span, Efficiency, Performance, Optimization</td>
<td>Better performance and efficiency in terms of make span</td>
<td>Complexity and long-time consumption</td>
<td>Finding solutions to large scale optimization problems</td>
<td>Complexity depends on the task to be scheduled</td>
<td>This is a greedy algorithm and pick the best job to allocate the CPU</td>
<td>It deals with a problem where the search space is large</td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>Make span, Efficiency, Performance, Optimization</td>
<td>Better performance and efficiency in terms of make span</td>
<td>Complexity and long-time consumption</td>
<td>Finding solutions to large scale optimization problems</td>
<td>Complexity depends on the task to be scheduled</td>
<td>This is a greedy algorithm and pick the best job to allocate the CPU</td>
<td>It deals with a problem where the search space is large</td>
</tr>
</tbody>
</table>
V. CONCLUSION

Cloud computing is one of the user oriented technologies in which user faces a group of virtualized resources of computer. Scheduling is one of the major significant activities of the process manager which take decision to select which of the process in the ready queue will be allocated to the CPU. There are several types of scheduling algorithms existing for taking decisions. In this paper, a number of available algorithms for job scheduling are discussed and also match up with each other. First Come First Serve algorithm has some drawbacks like processing time of every job must be well-known in advance and it is appropriate only for batch process. One of the main disadvantages of this scheme is that the more amount of average time. In Shorts Job First algorithms, extensive jobs may stay longer because it has to stay not only for jobs that are in the system at the time of its entrance, but also for all short jobs that are in the system at the time of its entrance. In Priority Algorithm only top priority jobs obtain chance to perform. Like first-come, first-served methods, Round-robin scheduling doesn't give any unique priority to additional essential tasks which means an imperative appeal doesn't get handled any quicker than other needs in the queue.

REFERENCES


