IMPLEMENTATION OF CLOUD SCHEDULING ALGORITHMS

K. Srikar¹, M. Akhil², Dr. V. Krishna Reddy³
¹²³ CSE, KLUniversity,
¹srikarkanna016@gmail.com

Abstract: Cloud computing is a trending technology which is rigorously increasing. During these days’ necessities of compute are also being increased to satisfy the requirements clients are depending on cloud services and are requesting cloud service provider, to handle these request made by clients. So cloud provider applies the scheduling for analyzing the request’s and handle them with robustness. Scheduling increases the performance efficiency of the cloud. scheduling is always a problem in cloud computing to handle remote requests, resources easily in order to reduce the load on machines so that they can yield the results quickly and dynamically.

Keywords: scheduling, types of scheduling, genetic algorithm (GA), particle swarm optimization (PSO), Particles, objective function.

1. Introduction

Cloud computing is a platform which enables us provide the remote resources like compute, network, storage over the network dynamically.

Cloud computing is model which provides us the basic resources anywhere and at any point of time by utilizing the resources of the basic data centers. cloud computing virtualizes these resources and provides the image for the users and user may not know where the resource is present. The resource request will be plenty at different levels of service different request will be asked by the user.

Scheduling in the area of cloud computing mainly enables for load balancing between the resources and handle the available resources continuously. Scheduling techniques are mainly classified into Static technique and Dynamic technique.

Static technique: this technique is limited to only particular type of application where the inputs and outputs are familiar to the user.

Dynamic technique: this technique can be applied to any type of applications in this we are familiar with the outputs and inputs because they will be continuously changing.

There is a special type of technique called heuristic technique which is like a trial and error technique and it yields many results for same problem.

A. Scheduling procedure

Scheduling can be done in mainly three steps and they are[3]:

- Find the available resource and list and store them based on its qualities.
- Find the resource from the list based on the analysis provided by the client
- Based on the request task targeted resource will be assigned to the client

These classifications can be completely based on service provider classification. scheduling can be done at service end, user end and middleware end.

2. Literature survey

There are many algorithms for performing scheduling in the environment of cloud to achieve maximum throughput or reduce the load on the resources, such as:

- First come first served
- Shortest job first
- Priority scheduling algorithm
- Min-Min algorithm
- Max-Min algorithm

And so many algorithms are used based upon the requirements and conditions

A. First come first served:

a. Start all tasks provided with execution times or not.
b. Task will be scheduled based on their arrivals.
c. If a new task is issued it will be placed after the tasks which came before.

B. Shortest job first:

a. Begin all tasks and provide execution times.
b. Find the shortest job in list and execute it first.
c. A[i]=[4,9,5,7,2,10], Sort(A); and it will provide them in an ascending order and now execute them.

C. Priority scheduling algorithm:

a. Begin all tasks and assign priority for every task.
b. Find the task with highest priority execute the task.
c. If a task with more priority occurs than the executing one it will be replaced with the new one.
d. Repeat the same until the end of task

Starvation is problem with the scheduling.

D. Min-Min algorithm:

If task queue is not empty

for (task in the set TS)

for (resources in the set RS)

Costij= executiontimeij + resourcetimeij

for (every value in C)

Search task Tk with the minimum and earliest complete time, relative Resource Rj
Assigning tasks $Tk$ to resource $Rj$
Removing task $Tk$ from TS
Update the execution time, return to 1
Exit.

**E. Max-Min algorithm:**
If task queue is not empty
for (task in the set TS)
  for(resources in the set RS)
    Cost$_{ij}$=execution time$_{ij}$ + resource time$_{ij}$
for (every value in C)
Search task $Tk$ with the maximum and earliest complete time, relative Resource $Rj$
Assigning tasks $Tk$ to resource $Rj$
Removing task $Tk$ from TS
Update the execution time, return to 1
Exit.

These are very good algorithms but these cannot be applied in every situation so we go for meta heuristic algorithms and try to solve them and find the optimal solution. These algorithms are also used in their respective domain and problem spaces where these are suitable.

## 3. Comparative Analysis

In here we are going to mainly discuss about Some metaheuristic algorithms like Genetic algorithm and Particle swarm optimization

**A. Genetic Algorithm**
This algorithm is developed by taking the inspiration of mother[4] nature and how it works. genetic algorithm mainly happens in six steps and they are selection, evaluation, selection, crossover, mutation, and repeat[5]

Algorithm:

Begin generate initial population(t) randomly calculate the fitness of every population by applying fitness function.
DO
Find parents from the population(t) Apply the function of crossover on parents creating population (t+1).
Apply operation of mutation for population(t+1) calculate the fit value of population’(t+1) repeat best child or particle found.

**Fitness:** it is a special kind of function which helps to determine how good the individual or particle.
**Figure 4.** plot of generation in genetic algorithm.

**Crossover:** in this stage we combine properties of the parents in order to get a better individual or particle, it is just like mating[7].

**Mutation:** this is very important it adds[8] randomness to the new individual so it will not be same as the parent or any in initial population.

**B. Particle Swarm Optimization**

This algorithm is nature based intelligence technique which is developed using the behavior of flock of birds searching for food or school of fishes escaping from predator[9].

Algorithm:

Initialize particles
Do
For all every particle
formulate fit value
If fit value is more[10] than the best fitness value(parentbest), set current value as new particlebest
Find and select the particle with the best fitness value of all the particle assign it as globalbest

For all every particle
Calculate particle vel and update the position
While maximum iteration is reached or minimum error condition is satisfied. Equations:

\[
\text{vel} = \text{vel}_i + a_1 \times \text{random} \times (\text{particlebest}_i - \text{pos}_i) + a_2 \times \text{random} \times (\text{globalbest} - \text{pos}_i) \\
\text{pos}_{i+1} = \text{pos}_i + \text{vel}_{i+1}
\]

m=inertia weight.

a1,a2 :acceleration coefficients random: random numbers

**5. Conclusion and Future Scope**

We have presented two scheduling algorithms PSO and GA and we analyzed their performance for the different population size and compared them. Based on the outcomes we can say that PSO yield better results and provide more efficiency to the cloud service provider.

In future research we try to compare our results with the other heuristic algorithms like Ant colony optimization(ACO) and Simulated Annealing(SA).

**References**

[1] Ashima mittal, Dr.pankaj Kumar, Existing computational models and heuristic techniques for cloud scheduling problems IEEE Computer society.


[3] Er. Shimpy, Mr. Jagandeep Sidhu Different scheduling algorithms in Different cloud environments,


<table>
<thead>
<tr>
<th>Initial population</th>
<th>GA(genetic algorithm)</th>
<th>PSO(particle swarm optimization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>-33.6901</td>
<td>-4.830</td>
</tr>
<tr>
<td>50</td>
<td>-57.7811</td>
<td>-4.765</td>
</tr>
<tr>
<td>100</td>
<td>-64.7981</td>
<td>-3.728</td>
</tr>
<tr>
<td>150</td>
<td>-61.3373</td>
<td>-3.831</td>
</tr>
<tr>
<td>200</td>
<td>-72.28956</td>
<td>-4.237</td>
</tr>
</tbody>
</table>


