Scheduling of Independent Tasks over Virtual Machines on Computational Cloud Environment

Shivam Bhagwani[1], Dr. Lokesh Kumar R[2], Akshat Shrivastava[3], Arpan Shrivastava[4]

[1,3,4] Student, Computer Science and Engineering, Vellore Institute of Technology, Tamil Nadu, India
[2]Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Tamil Nadu, India

---

Abstract - Cloud computing is emerging with the advancements in technology. Whether it is the need of storage, infrastructure, platform or software, major cloud providers are facilitating all of them to the needy. Not just that, cloud computing also plays a role in resource sharing, result aggregation, etc. with the help of distributed and parallel computing. But the main challenge that this field faces is the efficient use of resources granted for tasks. In order to regulate and utilize these resources efficiently, it is needed that the tasks and processes that the machines participate in, are scheduled and executed in a proper manner so that each and every machine is utilized to its maximum potential possible. This paper addresses this problem over virtual machines in computational cloud environment. Aim is to schedule and execute independent tasks over these machines using different algorithms available and compare their performances. This will be done with the help of cloud simulators.

Key Words: Cloud, Cloudlets, FCFS, SJF, Round Robin, PSO, Java, Cloudsim.

1. INTRODUCTION

It is known that resources are very necessary for a plan to successfully execute. Other than resources, it is very important to move ahead with them with a proper planning. This paper discusses some similar scenario in the cloud computing based environment. Cloud computing, as one of the newest and swiftly developing computer technologies, needs some similar resources and planning.

In cloud computing, resources are nothing but virtual machines, the CPUs which are part of those machines, memory and storage capacity of the executors on which tasks are to be executed, cloudlet schedulers, etc. All these resources are accessed by tasks which are nothing but cloudlets in the cloud environment.

Every task has its characteristics like task length, size, estimated memory required, estimated time required, etc. Similarly, these terms are used for cloudlets being cloudlet length, cloudlet size, memory required by the cloudlet, time required by the cloudlet, etc.

For the purpose of implementation, the paper uses two java frameworks namely CloudSim and WorkflowSim. CloudSim is well structured and robust set of packages which help in simulation and modelling of cloud computing infrastructure and services. WorkflowSim on the other hand is set of packages which are extended to provide implementation of planning algorithms before actually scheduling cloudlets for service on virtual machines.

Few components which play important role for this purpose include-

1. Cloudlet- This is similar to a task that has to be executed on cloud based environment with its own length that is similar to instruction length. Apart from this, it has properties such as that of image size and processing unit requirements.

2. Data Center- This is responsible for allocating core services at the level of infrastructure. This brings together all the configurations of resources which are going to execute cloudlets. Data Center also plays role in setting up of policies for memory and storage devices.

3. Data Center Broker-Acts as a mediator between user and service providers in a cloud ecosystem. With the help of Cloud Information Service (CIS) it recognizes suitable service providers for any task or set of tasks that are pending and are to be taken care of by some executor.

4. Host- This is a model for physical component on cloud based ecosystem. It has memory, a guided policy, bandwidth for virtual machines and of course list of processing elements.

5. VM Scheduling Policy- It is defined at two levels being Host Level at which specification for overall processing power is defined. At VM Level, the machine distributes its own processing power to tasks (or cloudlets) depending upon their characteristics. [1]

Using all the components listed above and including some more, this paper provides an analysis of four of the most popular algorithms in use. This set of algorithms includes

a. FCFS
b. SJF
c. Round Robin; and
d. Particle Swarm Optimization (A stochastic population-based algorithm)

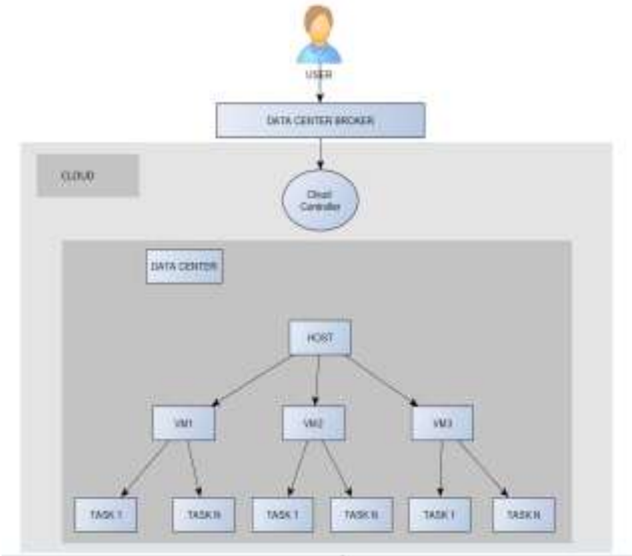

Figure 1: Cloudsim Architecture

This paper mainly focuses on comparison of three traditional algorithms with an advanced algorithm, make and produce an observation about how and to what extent PSO algorithm can deliver extra-ordinary performance which can't be achieved by the traditional ones discussed.

2. LITERATURE SURVEY

Many surveys have been conducted to come up with conclusions which all indicate an increment in the usage of cloud-based services over the last decade. Cloud services are considered as very essential. One of the key features of cloud computing is that it provides direct and remote access to many computing services. The examples of the services include servers, applications, network, etc. With this demand, it has been realised that scheduling of these services has become a very important issue. In year 2018, Anushree along with other co-authors completed a detailed analysis of variety of task scheduling algorithms. The algorithms include priority-based performance algorithm, template based genetic algorithm, hybrid multi objective PSO algorithm, intelligent water drop algorithm, improved genetic algorithm, etc. and found none to be satisfying on all parameters. [2]

Cloud computing is considered to be a computing skill that requires allocation of the computing resources as well as its accompanying services which are based on the pay per use model. In order to access the computers which are remotely located, scheduling is considered to be the most important task. Scheduling of the task is considered to be an NP complete problem. In order to achieve an optimum and better performance for the cloud resources, we need some successful and also some proficient methodologies of scheduling. The referenced paper discuses, a scheduling algorithm "Priority based Performance Improved Algorithm". This algorithm takes into consideration the priority of the meta-tasks of users. The resultant high priority meta-task is scheduled using the Min-Min algorithm whereas the normal priority one is scheduled using Max-Min algorithm. It concluded that the proposed algorithm is found to give minimum make span with better resource usage. [3]

Another publication by Gajera, Vatsal, Rishabh Gupta and Prasanta K Jana, an algorithm Min-Max which is widely used for normalization of data in the field of data mining is used as the basis for tuning the functionality to adjust with cloud computing. This came out to be named as Normalized Multi Objective Min-Min Max-Min Scheduling. It outperformed simple Min-Min and Max-Min algorithms. [4]

Toktam Ghafarian and Bahman Javadi in year 2015, proposed a cloud aware data intensive workflow scheduling system. As the current volunteer computing systems have the best infrastructure to execute high performance jobs however, they consume high computing power to run large data intensive scientific workflows. With an aim to solve the above problem by making a hybrid of the volunteer computing system and cloud resources possibility of enhancement in the efficiency in usage of these systems saw an increment. [5]

With the help of CloudSim, this process of analysis became very popular. Weighted Round Robin, Start Time Fair Queuing and Borrowed Virtual Time were among the many algorithms that got introduced and proposed. BVT proved to outperform the other two in a comparison provided by Jambigi, Murgesh V, Vinod Desai and Shrikanth Athanikar. [6]

Implementing algorithms for the purpose was not enough so few stepped forward to develop different frameworks. Li, Feng, Lin Zhang and Lei Ren introduced an idea of four layered model of task scheduling which included process layer, product layer, part layer and component layer. Genetic algorithms proved to perform well and many mutations and crossovers were tested for the purpose of queuing tasks. [7]

The scheduling of number of errands of processes, while handling with cloud assets, in a most beneficial way for example minimum computation time still happens to be an appealing examination region. Therefore, Sandeep Singh Brar and Sanjeev Rao illustrated the MaxMin calculation for scheduling work process undertakings that is centred on the thought of dependent and independent tasks and process independent tasks in parallel that legitimately gives benefit in minimizing computation time. [8]

3. SETUP AND EXPERIMENT

Before starting with the scheduling, planning is the step that has to be taken in order to prepare the environment for execution of task to take up the load of completion before the deadline. The difference between the current time and deadline guides the service provider to look up for a suitable scheduling heuristic and maintain cost and power.

A. VM Characteristics

Here, a very concise and to the point comparison of four different heuristic is presented. All these algorithms are tested over three differently capable virtual machines by tuning the number of CPUs embedded.

1. VM-1

   RAM: 512MBs

   MIPS: 1000

   Bandwidth: 1000

   CPU Cores: 1 (Single Core)

2. VM-2

   RAM: 512MBs

   MIPS: 1000

   Bandwidth: 1000

   CPU Cores: 2 (Dual Core)

3. VM-3

   RAM: 512MBs

   MIPS: 1000

   Bandwidth: 1000

   CPU Cores: 4 (Quad Core)

B. Pseudocodes for Algorithms Implemented

1. First Come First Serve

```
Step1: Put cloudlets in a queue data structure
Step2: First task waiting time will be 0.
Wt[0]=0;
Then calculating the whole waiting time
wt[i] =  bt[i-1] + wt[i-1];
Step3: Next is to calculate turn around time
Tat[i]=bt[i]+wt[i];
Step4: Calculate the average time
total_wt/n
total_tat/n
averagetime(processes,n,burst_time)
```

2. Shortest Job First

```
Step1: Create a heterogeneous cloud computing environment
Step2: Input all the required tasks and calculate the of tasks length for all the cloudlets
Step3: Sort them in ascending order using a sorting algorithm
Step4: Start with the first process and put other tasks in queue
For (process_id=0;process_id<n;process_id++)
Execute(cloudlet_process);
```

3. Round Robin

```
Step1: Initialise one array for keeping track of remaining execution time for cloudlets.
Step 2: Initialise another array for storing the waiting time of different cloudlets.
Step 3: Initialize time(t)=0
Step 4: Traverse through cloudlets while all of them are not processed. For any process:
        If remaining time > time slice
                t=t+time slice
                remaining time=-time slice
        Else (For the last cycle)
                t=t+ remaining time
                remaining time=0
```

4. Particle Swarm Optimization

```
Step 1: Define objective function f(x).
Step 2: Generate initial population of particles.
Step 3: Compute fitness of each particle and set particle best for individual particles.
Step 4: while (t<MaxGeneration) || (!stop)
        Select the GBest particle in swarm (minimum fitness value)
Step 5: Traverse through population
        Calculate fitness
        Calculate rank or position
Step 6: Traverse through population
        Assign new fitness value
        Assign new rank
Step 7: Find best particle.
```

C. Experiment

1)Running algorithms on VM-1

With single core processing capability, following results were obtained. It can be seen from the graph presented that PSO takes around 5th of the time taken by traditional algorithms.

Figure 1: Graph for make-span for different algorithms on VM-1

It is inferred that among standard algorithms, FCFS has emerged to be better performing as compared to other algorithms. SJF faces the issue of overhead incurred by the sorting of the cloudlets in increasing order of their length which is directly proportional to time of execution required by them. On the other hand, cloudlets in round robin sometimes wait for longer duration due to pre-emptive nature of the algorithm.

2) Running algorithms on VM-2

With dual core processing elements in the CPU, it is observed yet again that PSO brings out the best when the aim is to minimise the total execution time in the remote environment.

In this scenario, FCFS, SJF and Round Robin have performed almost as good as each other. There is nothing wrong in commenting that none of these have competed well enough with other two to outperform. Main reason for why Round Robin and SJF stay behind FCFS remains the same.

Figure 2: Graph for make-span for different algorithms on VM-2

3) Running algorithms on VM-3

Interesting outcomes were gained when these algorithms were tested on quad core CPUs which are highly prevalent these days in laptops, smartphones, computers and various other technical devices.

PSO doesn't show much improvement in this case but what is worth noting is the sudden increase of make span in round robin algorithm. This can be assigned to the collision of multiple CPU cores in order to process the same cloudlet. This is where some kind of lock should come into role for avoiding this collision.

Figure 3: Graph for make-span for different algorithms on VM-3

Another observation that attracted glances is that PSO still manages to perform as good as earlier not allowing the number of cores affecting its own results. It still completes managing and executing cloudlets in 5th of the time taken by SJF and FCFS.

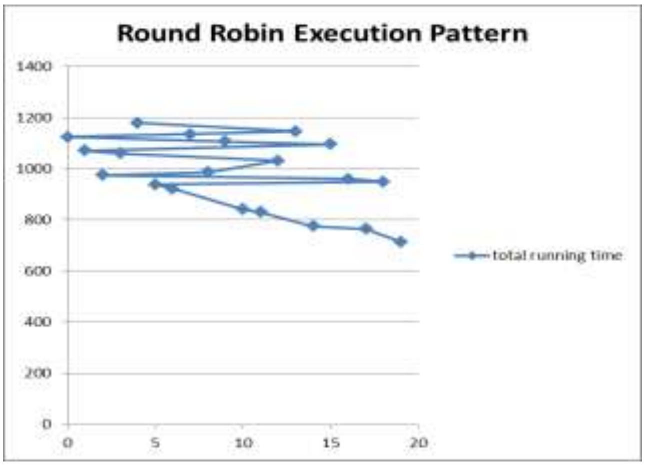

Figure 4: Round Robin execution pattern through line chart

4. RESULTS AND DISCUSSIONS

The table summarizes all the results that were gained from the experiments performed on heuristics over machines that happen to act and process differently on changing configurations.

	ALGORITHM			
NO OF CPU CORES	FCFS	SJF	ROUND ROBIN	PSO
1	50.05	52.55	54.27	13.93
2	25.08	26.41	27.48	4.16
4	13.09	13.84	34.26	3.1

TABLE 1: Overview of the results obtained through cross implementation of algorithms and Virtual Machines

5. REFRENCES

[1] Mondal, Choudhary. "Islam "Performance Analysis of VM Scheduling Algorithm of Cloudsim in Cloud Computing"." International Journal of Electronics & Communication Technology (2015): 49-53.

[2] Anushree, B., and VM Arul Xavier. "Comparative Analysis of Latest Task Scheduling Techniques in Cloud Computing environment." 2018 Second International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2018.

[3] Amalarethinam, DI George, and S. Kavitha. "Priority based performance improved algorithm for meta-task scheduling in cloud environment." 2017 2nd International Conference on Computing and Communications Technologies (ICCCT). IEEE, 2017.

[4] Gajera, Vatsal, Rishabh Gupta, and Prasanta K. Jana. "An effective multi-objective task scheduling algorithm using min-max normalization in cloud computing." 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). IEEE, 2016.

[5] Ghafarian, Toktam, and Bahman Javadi. "Cloud-aware data intensive workflow scheduling on volunteer computing systems." Future Generation Computer Systems 51 (2015): 87-97.

[6] Jambigi, Murgesh V., Vinod Desai, and Shrikanth Athanikar. "Comparative Analysis of different Algorithms for scheduling of tasks in Cloud Environments." 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). IEEE, 2018.

[7] Li, Feng, Lin Zhang, and Lei Ren. "A Production-Based Scheduling Model for Complex Products in Cloud Environment." 2017 5th International Conference on Enterprise Systems (ES). IEEE, 2017.

[8] Brar, Sandeep Singh, and Sanjeev Rao. "Optimizing workflow scheduling using Max-Min algorithm in cloud environment." International Journal of Computer Applications 124.4 (2015).