Comparative Analysis of Scheduling Algorithms of Cloudsim in Cloud Computing

Himani
Department of CSE
Guru Nanak Dev University, India

Harmanbir Singh Sidhu
Department of CSE
Guru Nanak Dev University, India

ABSTRACT

Cloud computing has emerged as a popular computing model to support on demand services and is rapidly becoming an important platform for scintific applications. It provide users with infrastructure, platform and software as amenity which is effortlessly accessible via Internet. It has a huge user group and has to deal with large number of task, so schedulling in cloud plays a vital role for task execution. In this paper, scheduling polices space-shared and time-shared are compared on the bases of some parametrs which are Task Profit, Task Penalty, Throughput and Net Gain. In our simulation results we shown that space-shared outperforms than time-shared policy.

Index Terms

Cloud Computing, Virtual Machine, Scheduling, CloudSim

1. INTRODUCTION

Now-a-days cloud computing rapidly gained the popularity of researcher, scholar and industries in recent years. It is the delivery of computing services and resources over the internet. There are many companies which are running in cloud computing enviornment and provide services and resouces companies like Amazon, IBM, Microsoft etc. The user can avail these services on the basis of pay per use manner at anytime from anywhere. Most of the reserchers try to define cloud computing with different perspectives. Among them, the definition provided by U.S. National Institute of Standards and Technology is a comparatively more relevant definition "The Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resource (e.g, networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. This definition not only defines cloud concept in general, but also explain essential characteristics of cloud computing delivery and deployment models.

Cloud computing offers everything as a service. Therefore, there are mainly three service which are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a Software as a Service (SaaS) model, a pre-made application, along with any required software, operating system, hardware, and network are provided. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications.

Many researchers have been carried out their work on the scheduling and allocation of the resources, task etc. in the cloud. Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system [2]. So scheduling is the leading issue in establishing Cloud computing systems. The main goal of these scheduling algorithm of task is to minimize the execution time and cost to achieve the maximum resource utilization. [3]. The motive of this paper is to focus on policies of scheduling task on virtual machine in cloud computing system. The rest of the sections are organized as follows. Section 2 Background of the work. Section 3 presents Scheduling Architecture of cloud computing. In Section 4 presents two scheduling policies and its comparison. Section 5 provide the implementation and analysis of respective polices. Section 6 concludes the paper with a summary of our future works.

2. BACKGROUND

Ke Liu [8] it presented a novel compromised-time-cost scheduling algorithm which considers the characteristics of cloud computing to accommodate instance-intensive cost-constrained workflows by compromising execution time and cost with user input enabled on the fly. The simulation has demonstrated that CTC (compromised-timecost) algorithm can achieve lower cost than others while meeting the user-designated deadline or reduce the mean execution time than others within the user designated execution cost. The tool used for simulation is SwinDeW-C (Swinburne Decentralized Workflow for Cloud). Abirami S.P and Shalini Ramanathan [9] mainly focus on the distribution of the resources among the requestors that will maximize the selected QoS parameters. This scheduling approach and the calculation of dynamic threshold value in the scheduler are carried out by considering both task and the resource. This improves the system throughput and the resource utilization. R. Santhosh [10] focus on providing a solution for online scheduling problem of real-time tasks using "Infrastructure as a Service" model offered by cloud computing. The real time tasks are scheduled pre-emptively with the intent of maximizing the total utility and efficiency. To minimize the response time and to improve the efficiency of the tasks. The tasks are migrated to another virtual machine whenever a task misses its deadline. This improves the overall system performance and maximizes the total utility. Xiaomin Zhua [11] The real-time controller and adaptive voltage controller work together and determine if an arriving task in the global queue can be admitted or not. Once the task is accepted, a voltage level will be assigned by the scheduler. Each node in the cluster maintains a local queue in which admitted tasks are queuing up for execution on the node. The local voltage controller in each node aims at minimizing the voltage levels for admitted tasks to reduce energy consumption.

3. SCHEDULING ARCHITECTURE

The main target of scheduling is to maximize the resource utilization and minimize processing time of the tasks. The scheduler should order the tasks so that balance between improving the quality of services and at the same time maintaining the efficiency and fairness among the tasks [4]. An efficient task scheduling strategy must goal to yield less response time so that the execution of submitted tasks takes place within a deadline as a consequences of this, tasks takes place and more number of tasks can be submitted to the cloud by the users which results in enhancement of the performance of the cloud system [5].

In Scheduling Architecture, Users submit their tasks to Data center Broker, this broker behaves like a dispatcher between user and Data center and helps to schedule task on virtual machines. In Data Centre there is number of host on which number of virtual machines are scheduled and on those VM task are scheduled according to the scheduling polices taken by Data Centre Broker. The number Data Centre Broker are similar to the number of users of cloud. The Data center Broker communicate with cloud controller and schedule the submitted tasks.

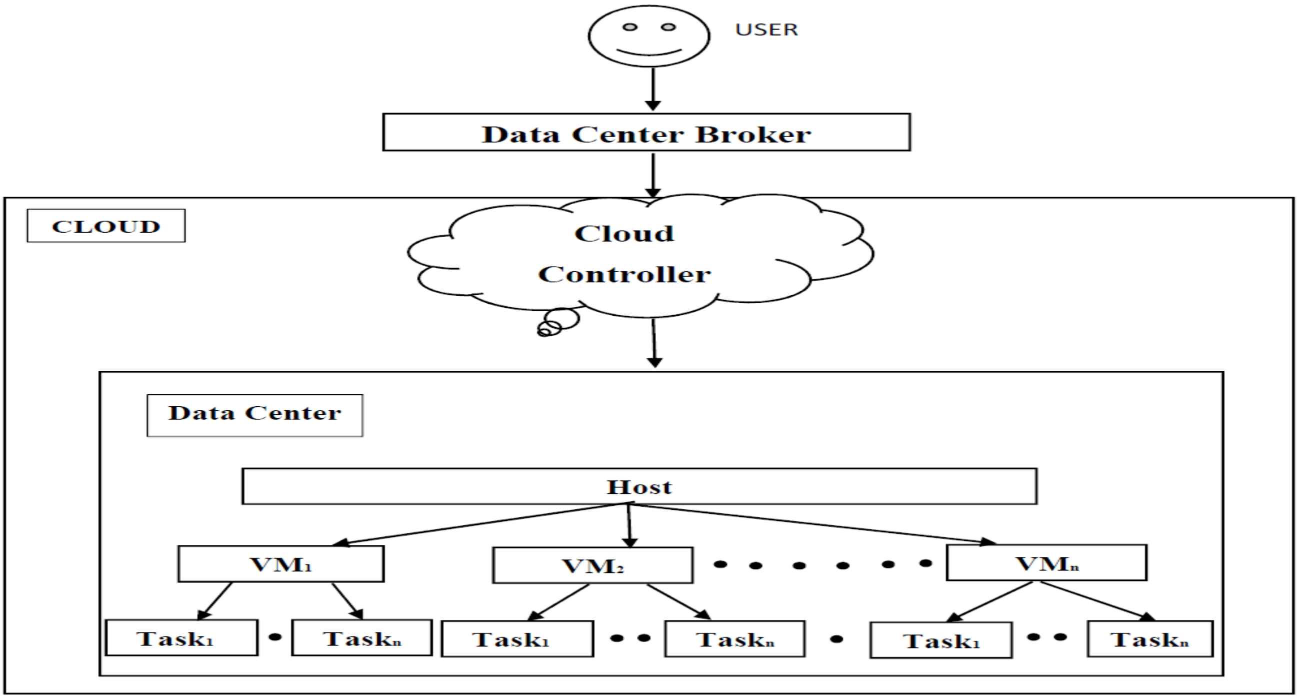

Figure 1. Scheduling Architecture

4. SCHEDULING POLICIES

There are two polices which are defined under this section these are Space-Shared scheduling policy and Time-Shared scheduling policy.

In space-shared scheduling policy it schedule one task on virtual machine at a given instance of a time and after its completion it schedule another task on virtual machine. This same policy is used to schedule the virtual machines on the host. This policy behave same as the first come first serve algorithm (FCFS) [6].

STEPS TO DEFINE SPACE-SHARED POLICY

Step 1:- Accepted tasks are arranged in a queue.

Step 2:- First task is schedule on the given virtual machine.

Step 3:- It completes first task and then take the next task from the queue.

Step 4:- If queue is empty it checks for new task.

Step 5:- Then repeats the step 1.

Step 6:- End

In Time-Shared scheduling policy it schedule all tasks on virtual machine at the same time. It shared the time among all tasks and schedule simultaneously on the virtual machine. This policy is also used to schedule the virtual machine on the host. The concept of round-robin (RR) scheduling algorithm [6] is used in this policy.

STEPS TO DEFINE TIME-SHARED POLICY

Step 1:- All accepted task are arranged under the queue.

Step 2:- Then schedule the task simultaneously on the virtual machine.

Step 3:- When queue is empty it checks for new task.

Step 4:- If new task arrives it schedule similarly as in the step 2.

Step 5:- End.

CloudSim implements the time-shared and space-shared scheduling policies. The difference between these two policies and their effect on the application performance, in Figure 2 show a simple scheduling technique. In which, a host with two CPU cores receives request for hosting two VMs, and each one requiring two cores and running four tasks units: t1, t2, t3 and t4 to be run in VM1, while t5, t6, t7, and t8 to be run in VM2.

Figure 2(a) presents a space-shared policy for both VMs and task units: as each VM entails two cores, only one VM can run at a given occasion of time. Therefore, VM2 can only be assigned to the core if VM1 finishes the execution of task units. The same happens for tasks hosted within the VM: as each task unit demands only one core, two of them run simultaneously, and the other two are queued until the completion of the earlier task units. In Figure 2(b), a space-shared policy is used for allocating VMs, but a time-shared policy is used for allocating individual task units within VM. Hence, during a VM lifetime, all the tasks assigned to it dynamically context switch until their completion. This allocation policy enables the task units to be scheduled at an earlier time, but significantly affecting the completion time of task units that are ahead the queue.

In Figure 2(c), a time-shared scheduling is used for VMs, and a space-shared one is used for task units. In this case, each VM receives a time slice of each processing core, and then slices are distributed to task units on space-shared basis. As the core is shared, the amount of processing power available to the VM is comparatively lesser than the aforementioned scenarios. As task unit assignment is space-shared, hence only one task can be allocated to each core, while others are queued in for future consideration.

Finally, in Figure 2(d) a time-shared allocation is applied for both VMs and task units. Hence, the processing power is concurrently shared by the VMs and the shares of each VM are concurrently divided among the task units assigned to each VM. In this case, there are no queues either for virtual machines or for task units. [7]

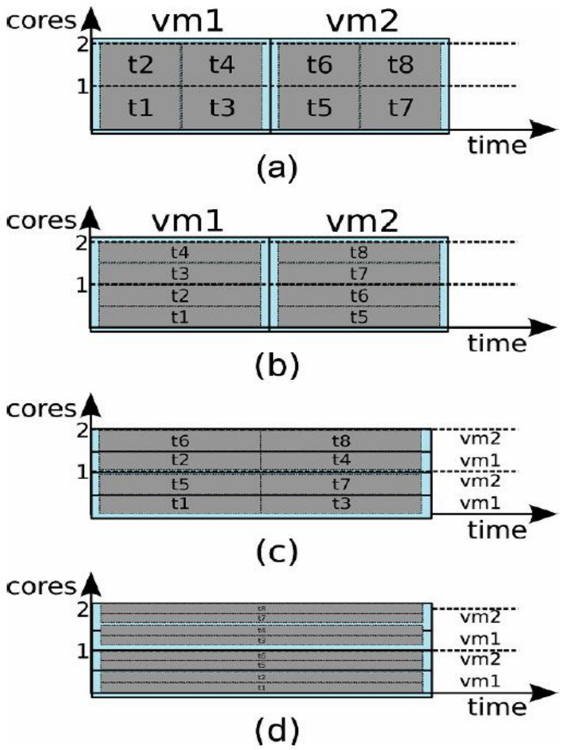

Figure 2. Effects of scheduling polices on task execution: (a) Space-shared for VMs and Tasks, (b) Space-share for VMs and Time-shared for tasks, (c) Time-shared for VMs, Space-shared for tasks, and (d) Time-shared for both VMs and Tasks [7]

5. SIMULATION RESULTS

In this section, we examine the practical abilities of Space-sahred and Time shared approach. We firstly extant the experimental situation and then compare the performances of Space-shared and Time-shared scheduling algorithm of cloud-sim.

A. *Scenario*

The performance measurement has been gathered by using the CloudSim [2] framework. The goal of this study is to show the ability of space-shared approach to meet deadlines in conditions where the Time Shared could not able to meet deadlines. To accomplish this, we recommend the following situation. We label identical datacenter composed of hosts. Each Processing element has a speed of 1000 MIPS. An available bandwidth of 10 Gbit/s. A storage capacity of 1 TB. A Random Access Memory (RAM) of 2048 MB.

We process 1000 cloudlets on the earlier demarcated datacenter using two policies: Space-Shared and Time-Shared approach. The lengths of cloudlets are equal and each cloudlet is given a randomly generated deadline where minimum value is half the number of cloudlets and maximum value is equal to number of cloudlets. Cost of each cloudlets is also depend upon the number of cloudlets and it is also randomly generated.

B. *Task Profit*

The number of tasks which have completed successfully before they meet the deadline. Space-shared approach is more efficient to finish tasks before the deadline arrives than Time-shared scheduling algorithm. We compare number of tasks completed by each approach for the given set of cloudlets. In figure 3. We can see the results of experiment.

On the x-axis it shows the number of cloudlets and y-axis indicates number of tasks completed successfully before deadline reaches. By examining the figure we can say that Time-shared algorithm inefficient to meet deadline before it comes.

We can notice that, in this case both approaches seems to be quiet indistinguishable but space-shared completes almost all tasks before it meets the deadline. By increasing the number of cloudlets and virtual machines we can observe the more difference and conclude that Space-shared approach is better than Time-shared approach.

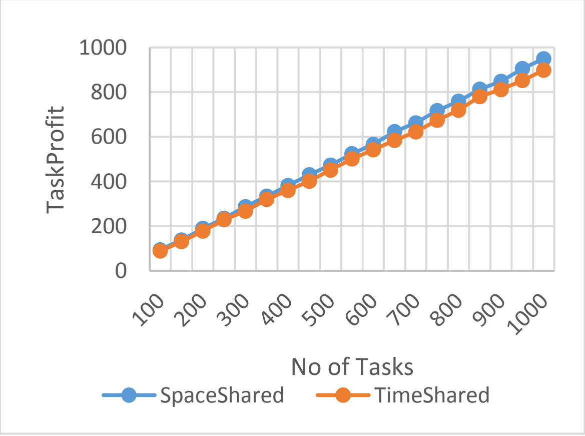

Figure 3. Comparison On the basis of Task Profit

C. Task Penalty

We can analyze result of another experiment from Figure 4. In this parameter, we annoyed to find task penalty of cloudlets i.e. number of cloudlets misses their deadline.

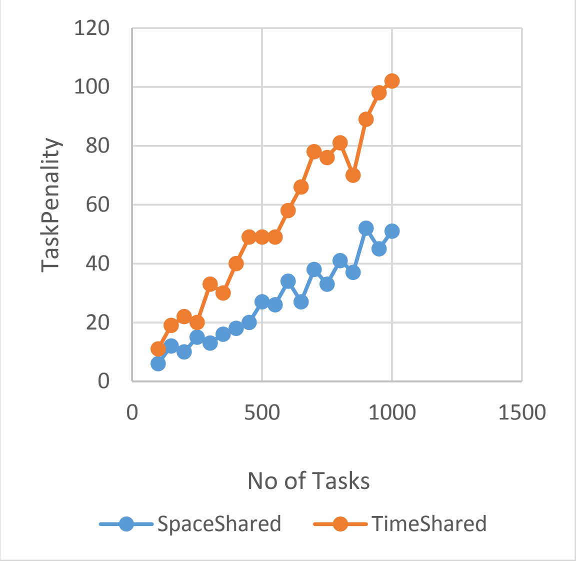

Figure 4. Comparison on the basis of Task penalty

The x-axis represents the number of cloudlets and y-axis defines missed deadline. Once again, it is notice that space-shared outperforms than other approach. It missed negligible number of deadline as compared Time-shared polices.

D. Throughput

Another parameter which shows space-shared approach is best among both polices. Throughput of tasks is the difference between earlier defined parameters. Figure 5 shows experimental results of all approaches in which x -axis indicates the throughput and y - axis represents the total number of cloudlets.

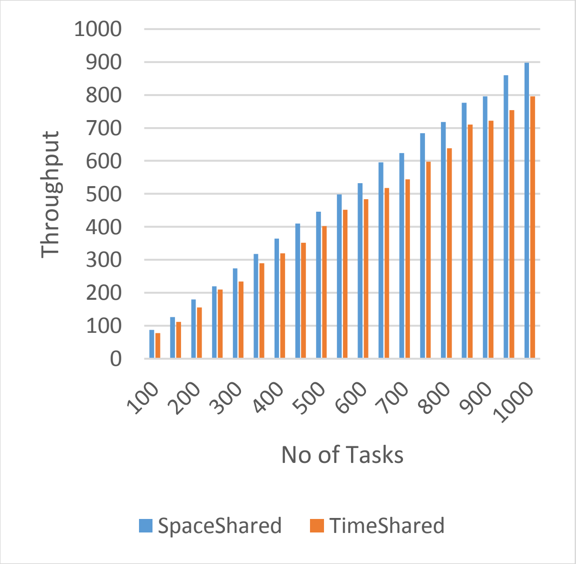

Figure 5. Comparison on the basis of Throughput

E. Net Gain

Space-shared is capable to process soft deadline cloudlets. Such cloudlets has penalties when they misses their deadline. To calculate the cost Net gain parameter is introduced which depends on the lateness of the cloudlets. So it is important to analyze the cost in order to measure the performance policies.

Figure 6. Results indicates that net gain in space-shared approach is maximum as compared to other policy.

In figure, x-axis shows number of cloudlets and y-axis shows the net gain of the cloudlets. Time-shared polices shows very less profit but for space-shared approach net gain increases as the number of cloudlets increases.

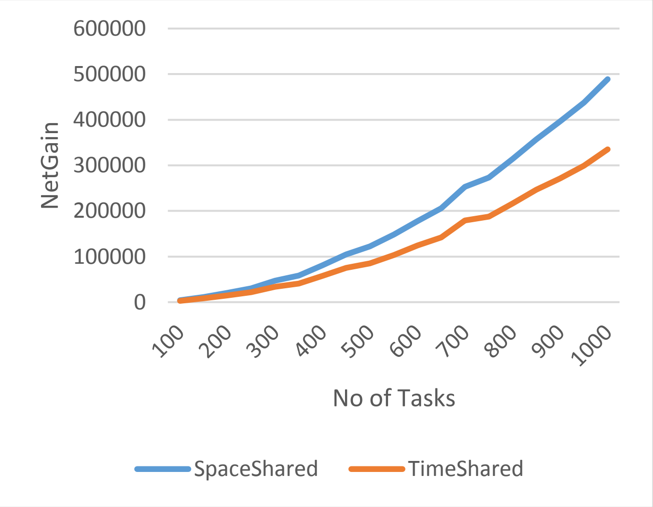

Figure 6. Comparison on the basis of Net Gain

6. CONCLUSION AND FUTURE SCOPE

This paper analyzes that space shared scheduling policy shows better results as compared to Time-shared scheduling policy. In task profit parameter shows little difference between both but task penalty shows more difference between them. When we increase the number of task the net gain between both shows more difference as it shows in 1000 tasks. In further research can be done for enhancing the efficiency scheduling policy algorithm for batter results by reducing scheduling complexities and improving computations for that we proposed a modified space shared policy in which deadline and cost parameter is used to evaluate the results.

7. REFERENCES

[1] The NIST definition of cloud computing, NIST special publication 800-145.

[2] Sandeep Tayal, “Tasks Scheduling Optimization for the cloud computing systems”, (IJAEST) International Journal Of Advanced Engineering Sciences and Technologies, Volume-5, No.2, p 11 – 15, 2011.

[3] Vijindra and Sudhir Shenai. A, “Survey of Scheduling Issues in Cloud Computing”, 2012, Elsevier Ltd.

[4] Isam Azawi Mohialdeen, “A Comparative Study of Scheduling Algorithms in Cloud Computing Environment” 2013 Science Publications.

[5] L.M.Vaquero, L.Rodero Merino, J.Caceres, and M.Lindner, “A break in the clouds: towards a cloud definition,” ACMSIG COMM Computer Communication Review, vol.39, no.1, pp.50–55,2008.

[6] Jiayin li, Meikang Qiu, “Online optimization for scheduling preemptively task on IaaS cloud systems”, Elsevier Inc 2012.

[7] Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities”.

[8] K. Liu; Y. Yang; J. Chen, X. Liu; D. Yuan; H. Jin, “A Compromised-Time- Cost Scheduling Algorithm in SwinDeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform”, International Journal of High Performance Computing Applications, vol.24, May,2010, Page no.4 445 456.

[9] Abirami S.P and Shalini Ramanathan, “ Linear Scheduling Strategy for Resource Allocation in Cloud Environment”, International Journal on Cloud Computing: Services and Architecture(IJCCSA),Vol.2, No.1,February 2012.

[10] R. Santhosh, T. Ravichandran, “Pre-emptive Scheduling of On-line Real Time Services with Task Migration for Cloud Computing”, International Conference on Pattern Recognition, IEEE, Feburary 2013.

[11] Xiaomin Zhua, Chuan Hea, Kenli Li, Xiao Qin, “Adaptive energy-efficient scheduling for realtime tasks on DVS-enabled heterogeneous clusters”, J.Parallel Distrib. Comput, SciVerse ScienceDirect, 2012, Elsevier Inc.