Analysis of Cloud Service Provisioning Policies using a GUI Simulator

Ashwin Semwal, Pradeep Rawat

***Abstract*— Cloud Services may be offered as public or private or combined. There is the demand of timely, repeatable, and controllable methodologies for evaluation of algorithms, applications, and policies before actual development of cloud products. Using simulation, we can study the behavior of them in cloud environment. Here I proposed a GUI based toolkit based on CloudSim library to simulate different cloud scenarios to find out the finish time taken by the SaaS modeler to run over the virtual machine using different resource provisioning algorithm i.e. time shared and space shared at each level. I follow the utility cloud computing based layered architecture. We use the different service broker policy at Application deployment configuration level to improve the performance. Variable configuration settings can easily be alter at any instance of time and simulation results provide the clue to identify the best service broker policy to setup the cloud configuration for the main cloud Model.**

***Index Terms*— Cloud Simulation, CloudSim, Provisioning Policies, Virtual Machine Allocation.**

I. Introduction

The offered Cloud Services classified as public or private or combined of both. These cloud architectures demand timely, repeatable, and controllable methodologies for evaluation of algorithms, applications, and policies before actual development of cloud services/products[1]. Cloud Simulation toolkit offer significant benefits to test cloud configurations over a controlled environment as it is free of cost and trust worthy before deploying it to the real cloud environment.

In Section 2, I have discussed some of the research issue like QoS, Flexible Mapping, Virtual Machine Allocation Techniques and Quality Provisioning Policy. Section 3, I have discussed about the CloudSim, the tool for simulation in cloud environment, and its working.

In Section 4, I have discussed working of a service broker provisioning policy of the CloudSim.

In Section 5 and Section 6, I have proposed GUI toolkit to simulate cloud architecture with some results. And finally I have concluded my work.

II. Research Issue

Here are some research issues discussed briefly regarding the context of paper.

A. QoS

Cloud Computing and service delivery model is all about the quality of service provided to cloud service consumer. Cloud Model is based on the pay as you go model. Consumers are billed as per their usage of data or time for which they are using the virtual machines. Any fault in configuration will affect the overall architecture performance of the system. Unlike many existing multi-objective optimization solutions, the optimization models that ultimately aim to optimize both resource-centric (utilization, availability, reliability) and user-centric (response time, budget spent, fairness) QoS targets need to be developed.

B. Flexible Mapping Services to Resources

With increased operating costs and energy requirements of composite systems, it becomes critical to maximize their efficiency, cost-effectiveness, and utilization. The process of mapping services to resources is a complex undertaking, as it requires the system to compute the best software and hardware configuration (system size and mix of resources) to ensure that QoS targets of services are achieved, while maximizing system efficiency and utilization. This process is further complicated by the uncertain behavior of resources and services.

C. Virtual Machine Allocation and Quality provisioning Policies

One implication of Cloud platforms is the ability to dynamically adapt (scale-up or scale-down) the amount of resources provisioned to an application in order to attend variations in demand that are either predictable, and occur due to access patterns observed during the day and during the night; or unexpected, and occurring due to a subtle increase in the popularity of the application service. This capability needs a proper allocation of virtual machine to a service and a quality of provisioning policies (depends on broker) to setup the dynamic configuration.

To enable the new mobile cloud application model, many challenges exist in different areas, including data replication, consistency, transaction management, cache management, optimal cost-effective execution in heterogeneous computing environments [2].

III. Cloud-Sim

Cloud-Sim is developed on the top of the Grid-Sim. Cloud-Sim is a console based Toolkit to configure and test different cloud configurations. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic

Manuscript received March 20, 2014.

Ashwin Semwal, Computer Science and Engineering, DIT, Dehradun, Uttarakhand, India ,Mob: 9808183987,

application provisioning techniques that can be extended with ease and limited efforts.

Application users: There is the requirement of autonomous entities to act as traffic generators and behavior needs to be configurable.

Internet: It is introduced to model the realistically data transmission across Internet with network delays and bandwidth restrictions.

A. Layered Cloudsim Architecture

The CloudSim simulation layer provides support for modeling and simulation of virtualized Cloud-based data center environments including dedicated management interfaces for virtual machines (VMs), memory, storage, and bandwidth [3].

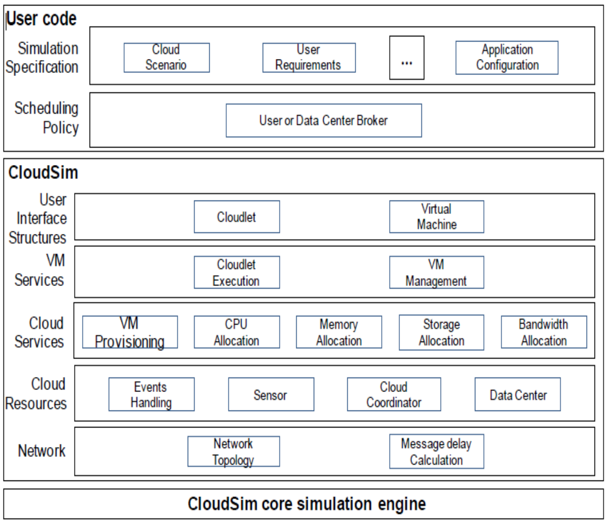

Figure 1 Layered CloudSim Architecture

The fundamental issues such as provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state are handled by this layer. A Cloud provider, who wants to study the efficiency of different policies in allocating its hosts to VMs (VM provisioning), would need to implement their strategies at this layer. Such implementation can be done by programmatically extending the core VM provisioning functionality.

B. Some CloudSim Components

1) Cloudlet

This class models the Cloud-based application services (such as content delivery, social networking, and business workflow). CloudSim orchestrates the complexity of an application in terms of its computational requirements. Every application service has a pre-assigned instruction length and data transfer (both pre and post fetches) overhead that it needs to undertake during its life-cycle. This class can also be extended to support modeling of other performance and composition metrics for applications such as transactions in database-oriented applications.

2) CloudletScheduler

This abstract class is extended by implementation of different policies that determine the share of processing power among Cloudlets in a virtual machine. As described previously, two types of provisioning policies are offered:
i).Space- Shared (CloudetSchedulerSpaceShared).
ii).Time- Shared (CloudletSchedulerTimeShared).

3) DataCenter

This class models the core infrastructure level services (hardware) that are offered by Cloud providers (Amazon, Azure, App Engine). It encapsulates a set of compute hosts that can either be homogeneous or heterogeneous with respect to their hardware configurations (memory, cores, capacity, and storage). Furthermore, every Datacenter component instantiates a generalized application provisioning component that implements a set of policies for allocating bandwidth, memory, and storage devices to hosts and VMs.

4) Vm

This class models a virtual machine, which is managed and hosted by a Cloud host component. Every VM component has access to a component that stores the following characteristics related to a VM: accessible memory, processor, storage size, and the VM's internal provisioning policy that is extended from an abstract component called the CloudletScheduler.

5) VmAllocationPolicy

This abstract class represents a provisioning policy that a VM Monitor utilizes for allocating VMs to Hosts. The chief functionality of the VmAllocationPolicy is to select available host in a data center that meets the memory, storage, and availability requirement for a VM deployment.

IV. Service Provisioning Policies

CloudSim uses two different provisioning policies i.e. Space Shared and Time Shared provisioning policies to allocate virtual machine over a datacenter host.

In space shared T1,T2,T3….T8 tasks are allocated using space sharing scheme and second task will always get allocated after the first task is completed until then the second task has to wait for its execution.

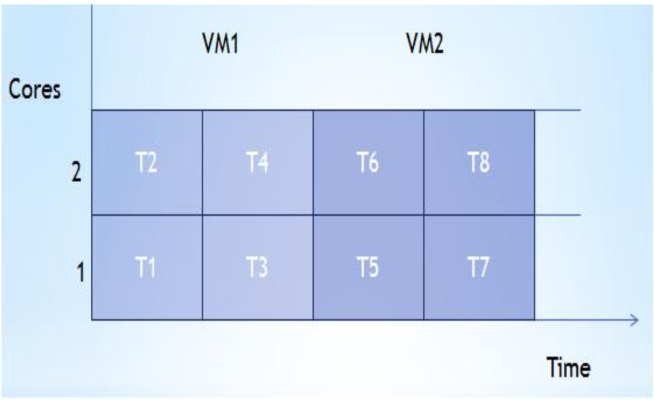

Figure 2 Space Shared Policy

In time shared T1,T2,T3….T8 tasks are allocated all at once and are timely executed over memory. In this the execution start all at once (depending on memory) and all tasks are simultaneously executed over time bounded scenario.

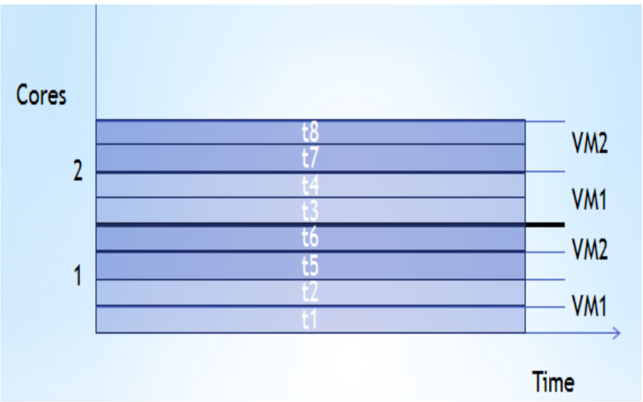

Figure 3 Time Shared Policy

V. Cloud Simulator GUI

Here I proposed a GUI cloud simulator developed on top of CloudSim discussed earlier. This GUI Cloud Simulator proves best to analyze the different provisioning policies and different configuration settings. Some of the main component of GUI model is discussed here.

A. *Variable DataCenter Allocation*

Data center configuration setting can be easily set using simple interface in which MIPS, STORAGE and CPU CORES can easily be configured.

B. *Variable Virtual Machine Configuration*

Virtual machine configuration like RAM, STORAGE, MIPS, and PROCESSOR can be altered and new virtual machine can be entered from this interface.

C. *Variable Cloudlets Distribution*

Task length can be altered and initialized from this interface also provisioning policy can be decided for the cloudlets allocation.

D. *Hybrid Provisioning Policies*

Hybrid provisioning policies can be initialized by allocating relevant different policies for virtual machine and cloudlet allocation during simulation.

E. *Graphical allocation representation & Result*

Result are shown in a tabular format showing the total executing time taken for the configured model and it also provide a graphical execution summary for each cloudlet running on virtual machines.

GUI and Ability to represent simulation results, user can configure the simulation with high level of details using the GUI. It makes easy to do the simulation experiments and to do it in repeatable manner at any instant of time. Using the GUI introduced here, we can also compare the utilization of different virtual machines and cloudlets running on them based on their final finishing time during the simulation using a graphical representation.

VI. Simulation Results Using GUI

For each Scenario I take this initial configuration for datacenter as:

Table 1. Data Center Configuration

Host ID	Storage (in Mb)	MIPS	CPU Core	Bandwidth	RAM (in Mb)
0	100000000	10000	4	1000000	204800

Table 2. Virtual Machine Configuration

ID	Storage (in Mb)	MIPS	CPU Cores	Bandwidth	RAM in mb
0	10000	1000	2	1000	1024
1	10000	1000	2	1000	1024

Table 3. Cloudlet Configuration

Host ID	Length	In/out	CPU Cores
0..7	250000	300	1

After this configuration we choose both the provisioning policies for Vm one by one and results are simulated as follows:

Table 4. Results for Space Shared Provisioning policy

Vm ID	Cloudlet handled	Finish Time
0	4	249.5658
1	4	501.0009
		Avg response time: 375.28ms

In Space Shared the second virtual machine remains halted until the first virtual machine finishes its execution.

Table 5. Result for Time Shared Provisioning Policy

Vm ID	Cloudlet handled	Finish Time
0	4	249.5648
1	4	251.4351
		Avg response time : 250.50045ms

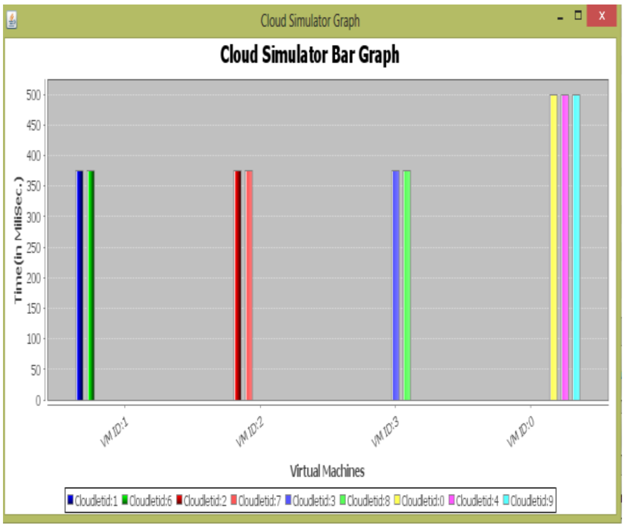

Figure 4 Allocation Graph by Simulator

In Time Shared the processing power is concurrently shared by the VMs and the shares of each VM are simultaneously divided among its task units. In this case, there are no queuing delays associated with task units.

For a time shared configuration, 9 cloudlets distribution over 4 different virtual machines in the simulator is represented graphically in figure 4.

The Graphical result shows the overall time taken by each cloudlet and the virtual machine on which it is allocated by the provisional policy. The graph can easily be read and use to deduce the allocation techniques and average response time.

VII. Conclusion

With the advancement of Cloud technologies rapidly, there is a new need for simulation tools to study and analyze the benefits of the technology and how to apply the technology to the large-scale applications. Tool based analysis of cloud computing environment is the first step to deploy our application on real cloud computing environment e.g. Google Cloud Storage[4], Amazon EC2[5]. In this paper I use variable data center configuration, and setup the simulation scenarios and identify the best service provisional policy at main cloud configuration for real deployment of application. I also demonstrated how this GUI Cloud-Simulator can be used to model and evaluate a real world problem through a simple configuration implementation. Simulator results and Cloudlet distribution graph is also presented. We have illustrated how the simulator can be used to effectively identify overall usage patterns and how such usage patterns affect data centers hosting the application. This Simulator is helpful in understanding the different configuration scenarios by analyzing the statistical output results and graphs.

Acknowledgment

I thank Mr. Pradeep Rawat, Assistant Professor who guides me through my research-work.

References

[1] R. Buyya, C. S. Yeo, and S. Venugopal, “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities”, Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008, IEEE CS Press, Los Alamitos, CA, USA), Sept. 25-27, 2008, Dalian, China.

[2] Dejan Kovachev, Yiwei Cao and Ralf Klamma Mobile Cloud Computing: A Comparison of Application Models Information Systems & Database Technologies RWTH Aachen University, Aachen Germany 2010-2011

[3] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities,” Proc. of the 7thHigh Performance Computing and Simulation Conference (HPCS09), IEEE Computer Society, June 2009.

[4] Google App Engine, http://appengine.google.com [18 April 2010].

[5] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/ [18 April 2010].

[6] Ashwin Semwal and Pradeep Rawat, “Performance evaluation of cloud application with constant data center configuration and variable service broker policy using CloudSim” IJERSTE, [Vol. 3 Issue 1, January-2014].

Ashwin Semwal, student of M.TECH (Computer Science and Engineering) in Dehradun Institute of Technology. He has done his B.TECH form DBIT Dehradun. He also works as a software trainer at HCLcdc. He finds his interest in Cloud Technology and Networks Engineering and has a publication in International Journals.

Mr. Pradeep Rawat, Assistant professor (CSE) Dit Dehradun, He has done his graduation from Kumaon Engineering College, Dwarahat. He is a technology enthusiast and has many publications in International Journals.