2013 Ninth International Conference on Computational Intelligence and Security

Using CloudSim to Model and Simulate Cloud Computing Environment

Wang Long, Lan Yuqing and Xia Qingxin
Department of Computer Science and Engineering
Beihang University
Beijing, China
Email: lanyuqing@buaa.edu.cn,wanglong,xiaqingxin@cse.buaa.edu.cn

***Abstract*—Cloud computing will be a major technology in the development of the future Internet of Services. Service providers want to remove the bottle neck of the cloud computing system in order to satisfy user requirement. And in order to save energy consumption they also need to apply new energy-efficiency mechanism and observe its effects on a datacenter. As it is difficult to test new mechanism in real cloud computing environment and researchers often cannot reach the real cloud environment, simulation to model the mechanism and evaluate the results is necessary. Simulating a datacenter avoids spending time and effort to configure a real testing environment. Moreover, as real machines are not used for testing purposes, their computational power can be allocated to profit-making applications. This paper introduces a simulation framework called CloudSim which provides simulation, power to manage services and modeling of cloud infrastructure. And we have also discussed about how to extend it to simulation your own mechanism in cloud computing.**

***Keywords*-cloudsim; cloud computing; simulation; energy effective; green computing**

I. Introduction

Cloud computing[1] will be a major technology in the development of the future Internet of Services, and it delivers infrastructure, platform, and software that are made available as subscription-based services in a pay-as-you-go model to customers. Data is neither stored on the local hard drive of your computer, nor on servers that are down in the basement of your company. Instead it is out in the cloud. The infrastructure is outside of your organization and you access the applications, the infrastructure, and all those services typically through the web-based interface. Customers can accept the cloud system everywhere and receive sufficient computing environments. These cloud service are divided into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). And, cloud computing has features of the distributed computing environment such as grid computing system. Therefore, if the cloud user pays appropriate costs, it is able to create a computing environment which is nearly infinite.

In order to provide cloud service, companies including Google, Amazon, Facebook, and Microsoft have multiple datacenters with thousands to millions of geographically distributed servers. The cloud provider has to offer a cloud environment which responds to the needs of the cloud users. Thus, the cloud service provider needs to check the bottleneck of services. Customer will not be satisfied with the services, if there is a bottleneck while providing various services. Also, the provide may suffer from a Service Level Agreement (SLA) problem, which is negotiated contract between a customer and a service provider that clarifies all service features that are to be provided, and consequently the policies that are to be taken in this way. Therefore, they need some tests to remove the bottleneck.

On the other hand, the servers in datacenters comprise a large portion of the overall IT energy cost, with networking infrastructure, such as switches and routers. A recently study revealed that the total energy consumed by all datacenters in the world increased 56 percent from 2005 to 2010[2]. Energy efficiency will remain a central goal in the future as the price of brown energy. Energy efficient solutions, which range from utilizing green energy source such as solar and wind power to optimizing hardware, software, and system design for energy efficiency, will thus continue to be important. As it is difficult to deploy a new energy-efficiency mechanism and observe its effects on a datacenter in a controlled and a repeatable manner. Researchers often use simulation to model the mechanism and evaluate the results. Simulating a datacenter avoids spending time and effort to configure a real testing environment. Moreover, as real machines are not used for testing purposes, their computational power can be allocated to profit-making applications.

CloudSim is the simulation program for the above two purpose[3]. It enables seamless modeling, simulation, and experimentation of cloud computing environment and the application services. And it also shows the result of the time, power, and traffic consumption. This paper will introduce the CloudSim simulator including its architecture, and how to use it to modeling the cloud environment.

This paper is organized as follows. In section 2 we discuss the related work. In section 3 we present the detail about CloudSim simulator. We introduce VM scheduling, provisioning and selection policy in cloud computing environment in section 4. We conclude the paper and discuss future work in section 5.

978-1-4799-2548-3/13 $31.00 © 2013 IEEE
DOI 10.1109/CIS.2013.75

II. Related Work

Simulation enables the evaluation of mechanisms that might not be implementable without a substantial investment, such as analyzing the cost and benefits of adding and managing electricity form photovoltaic. Many simulations platforms are available for evaluating the energy efficiency of datacenter environments. Each has its own design goals and used different resource, workload, queuing, and power models to accomplish those goals. For example, GreenCloud [4] is helpful for comparing different power-saving methods, such as dynamic voltage and frequency scaling (DVFS) and dynamic power management (DPM).

Developed by researchers at Penn State University, MDCSim[5] evaluates energy efficiency by quantifying datacenter power consumption on a per-server basis and by estimating the response times of datacenters jobs. MDCSim uses a simple utilization-based power model for servers: $P = P_{idle} + CPU_{util} \times (P_{peak} - P_{idle})$,where P is total power consumption, P_{peak} is peak power consumption, P_{idle} is idle power consumption, and CPU_{util} is CPU utilization.

Stochastic queuing simulation (SQS)[6], a methodology created by University of Michigan researchers, improves on MDCSim by estimating average datacenter service job response times in more detail. It models a datacenters as a pool of servers, each with an M/G/k queue.

The Data Centre Specialist Group Simulator (DCSG Simulator)[7], developed by British Computer Society in conjunction with Romonet Ltd., analyzes different datacenter layouts and estimates system cost based on the set of elements to be used and their relative placement. It conducts both infrastructure and IT device analysis. Barcelona Tech researchers developed EEFSim[8] to estimate the energy efficiency of a virtualized datacenters. It is designed to evaluated different scheduling, consolidation, and migration algorithms for VMs and show how these algorithms affect power efficiency.

GDCSim[9], a green datacenter simulator developed by researchers at Arizona State University and Xerox Research Center Webster, estimates the energy efficiency and thermal properties of datacenters as a function of floor plan, power management, and scheduling policies. Each server is represented by a process queue associated with each of its cores. The queuing model and scheduling algorithm are user-specified.

In this paper, we will describe CloudSim simulator, which is developed by the researchers in Australia and Brazil. It can simulation of Cloud computing scenarios, and provides basic classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system.

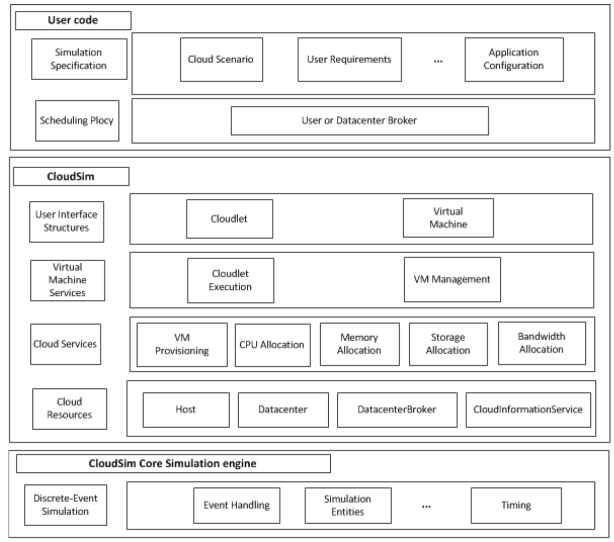

Figure 1. The CloudSim architecture

III. CloudSim Modeling

CloudSim[3] is a simulating program from CLOUDS[1] lab in University of Melbourne for cloud computing. It is developed in java platform including the pre-developed modules such as SimJava and GridSim. The authors have mentioned that users could analyze specific system problems through CloudSim, without considering the low level details related to Cloud-Based infrastructures and services.

A. CloudSim Architecture

Figure 1 shows the architecture of Cloud, the detail of each component can be found in article[3]. The layered architecture of the CloudSim is consisting of the simulation engine, cloud services and source code. CloudSim bills itself as an extensible simulation toolkit that enables modeling and simulation of cloud computing systems and application provisioning environments. Therefore, using CloudSim to simulate the cloud computing environment, users do not need to consider the low level of CloudSim.

User code column has the application configuration and the data center with the scheduling policies, as that the user code has the simulation specifications and the policy for the scheduling. As the cloud computing is a rapidly evolving research area, there is lack of defined standards, tools and methods that can efficiently tackle the infrastructure and application level complexities. Hence in the future there would be a number of researchers both in academia and industry towards defining core algorithms, policies, application benchmarking based on execution contexts. By extending the basic functionalities already exposed by CloudSim, researchers would be able to perform tests based on specific scenarios and configurations.

[1]http://www.cloudbus.org/cloudsim/

The basic scenario of CloudSim can be described as follows: that Datacenter has one or many Host, and each host has one or many Virtual Machines (VMs). Each VM deals with many cloudlets which are the units of cloud service. VM is assigned several cloudlets and processed them using scheduling policy such as time-sharing and space-sharing. CloudSim prints the result of processes. It shows a time consumption of each cloudlet. A cloudlet is a unit of cloud services; it has a set of consumption of clouds service. It specifies a program instruction length program size, output data size, and etc.

B. Modeling the cloud

The infrastructure-level services (IaaS) related to the clouds can be simulated by extending the Datacenter entity of CloudSim. The datacenter entity manages a number of host entities. The hosts are assigned to one or more VMs based on a VM allocation policy that should be defined by the cloud service provider. The VM allocation policy will be described in section 3.3.

A Datacenter can manage several hosts that in turn manage VMs during their life cycles. Host is a CloudSim component that represents a physical computing server in a Cloud: it is assigned a pre-configured processing capability (expressed in millions of instructions per seconds – MIPS), memory, storage, and a provisioning policy for allocating processing cores to virtual machines. The host component implements interfaces that support modeling and simulation of both single-core and multi-core nodes. To simulate a datacenter, you can just create a class which extends from the Class Datacenter, and the same as Host.

A Cloudlet can model 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 is needs to undertake during its life-cycle.

C. Modeling the VM Allocation

VM allocation (provisioning) is the process of creating VM instances on hosts that match the critical characteristics (storage, memory), configurations (software environment), and requirements (availability zone) of the SaaS provider. CloudSim supports the development of custom application service models that can be deployed within a VM instance and its users are required to extend the core cloudlet object for implementing their application services. Once an application service id defined and modeled, it is assigned to one or more pre-instantiated VMs through a service specific allocation policy. Allocation of application-specific VMs to Hosts in a Cloud-based data center is the responsibility of a Virtual Machine Allocation controller component (called VmAllocationPolicy). This component exposes a number of custom methods for researchers and developers that aid in implementation of new policies based on optimizations goals. By default, VmAllocationPolicy implements a straightforward policy that allocates VMs to the Host in First-Come-First-Serve (FCFS) basis. To implement your own VM Allocation policy, just extends the class VmAllocationPolicy.

D. Modeling the Dynamic Workloads

To make a simulation-based evaluation applicable, it is important to conduct experiments using workload traces form a real system. In CloudSim 3.0, the authors used data provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab[10]. They have used the data on the CPU utilization by more than a thousand VMs form servers located at more than 500 places around the world. The interval of utilization measurements is 5 minutes and each traced file have 288 lines, therefore, each one represent a VMs CPU utilization about 24 hours.

In the example which contained in the CloudSim originally, these workloads file are used to instance cloudlet which models cloud-based application services. The CloudSim simulator assigns the cloudlet to VMs sequentially.

The CloudSim also model the workloads randomly. The UtilizationModelStochastic class implements a model, according to which a Cloudlet generates random CPU utilization every time frame.

E. Modeling the Data Center Power Consumption

Power consumption by computing nodes in data centers is mostly determined by the CPU, memory, disk storage, power supplies and cooling systems. Recent studies have shown that the power consumption by servers can be accurately described by a linear relationship between the power consumption and CPU utilization, even when Dynamic Voltage and Frequency Scaling (DVFS) is applied.

CloudSim provide several Power models, such as PowerModelSqrt, PowerModelLinear, PowerModelSquare and PowerModelCubic. These power models are all based on the linear relationship between power consumption and CPU utilization. As to analysis and build the precise power consumption model is a complex research problem. Therefore, instead of using an analytical model of power consumption by a server, CloudSim also provide power model called PowerModelSpecPower, which utilize real data on power consumption provided by the results of SPECpower benchmark.

For example, Table I shows the relationship between the power consumption and CPU utilization of two servers (PowerEdge C6105 [11] and PowerEdge C6145[12]).

In our own simulation experiment, we can use self-defined power model by extend the class PowerModel, linear power model by extend the class PowerModelLinear and SPCE-power model by extend the class PowerModelSpecPower.

Table I
POWER CONSUMTPTION AT DIFFERENT LOAD LEVELS IN WATTS

Server(%)	0	10	20	30	40	50	60	70	80	90	100
C6105	106	187	229	266	307	350	408	458	501	538	573
C6145	196	362	437	510	595	690	790	883	962	1038	1105

The utilization of the CPU may change over time due to the workload variability. Thus, the CPU utilization is a function of time and is represented as $u(t)$. Therefore, the total energy consumption by a physical node (E) can be defined as an integral of the power consumption function over a period of time as shown in equation (1).

$$E = \int_{t_0}^{t_1} P(u(t))\, dt \tag{1}$$

And the total power consumption of cloud datacenter can be defined as a sum of power consumption function of all nodes as shown in equation (2). Where N is the number of the physical node in the datacenter,$u(t_i)$ is the CPU utilization by the physical node i.

$$E_{datacenter} = \sum_{i=0}^{N} E_i = \sum_{i=0}^{N} \int_{t_0}^{t_1} P(u(t_i))\, dt \tag{2}$$

IV. Policy and Algorithms

A. VM Scheduling Policy

CloudSim models scheduling of CPU resources at two levels: Host and VM.

At Host level, the host shares fractions of each processor element (PE) to each VM running on it. Because resources are shared among VMs, this scheduler is called VmScheduler. The scheduler a host uses is a parameter of the Host constructor.

In the VM level, each virtual machine divides the resources received from the host among Cloudlets running on it. Because in this level resources are shared among Cloudlets, this scheduler is called CloudletScheduler. The scheduler a VM uses is a parameter of its constructor.

In both levels, there are two default policies available: the first policy, xSpaceShared (x stands for VmScheduler or CloudletScheduler), required PEs by Cloudlets/VMs are exclusively allocated. It means that if there are more running elements (VMs or Cloudlets) than available PEs, the last elements to arrive wait on a queue until enough resources are free. In the second policy, xTimeShared, fraction of available PEs are shared among running elements, and all the elements run simultaneously.

Policies for VM scheduling and Cloudlet scheduling can be used in any combination. For example, researchers can use VmSchedulerTimeShared and CloudletSchedulerSpaceShared, or researchers can use VmSchedulerTimeShared and CloudletSchedulerTimeShared. It is possible even having a host running VMs with different Cloudlet scheduling policies, or a data center with hosts with different VM Scheduling policies.

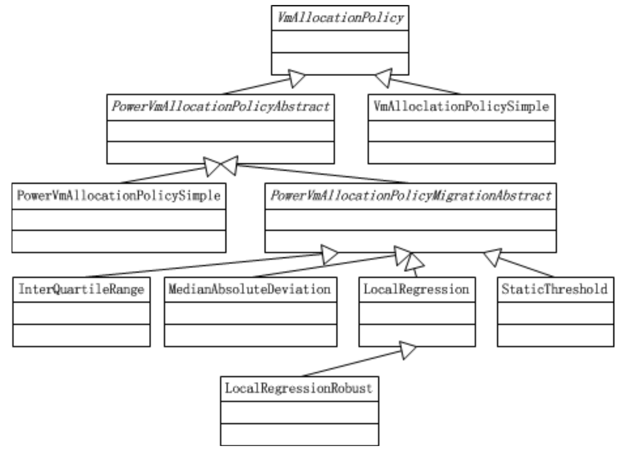

Figure 2. The provisioning policy

To define your own policy, you have to extend either VmScheduler or CloudletScheduler, create the methods for deciding sharing of PEs and pass the new class during construction of the relevant object. For example, extend VmScheduler and pass the object to the host. Or extend CloudletScheduler and pass the object to the VM.

B. VM Provisioning Policy

The provisioning problem consists of defining, among the available hosts in the data center, which one should receive a new machine requested by a user. Provisioning of hosts to VMs in data centers follows a simple strategy where the host with less running VMs receives the next VM. This behavior is defined in the VmAllocationPolicySimple class. To change this behavior, extend VmAllocationPolicy to define the new provisioning behavior, and pass this object in the initialization of Datacenter. The provisioning policy in the CloudSim 3.0 can be described in figure 2.

VmAllocationPolicy is an abstract class that represents the provisioning policy of hosts to virtual machines in a datacenter. PowerVmAllocationPolicyAbstract is an abstract class which defined a power-aware VM allocation policy. PowerVmAllocationPolicyMigrationAbstract is an abstract class which defined a power-aware VM allocation policy that dynamically optimizes the VM allocation using migration. The other classes, which defined some different policy of power-aware VM allocation policy, are all extended from PowerVmAllocationPolicyMigrationAbstract or PowerVmAllocationPolicyAbstract. Researchers can also implement dynamic VM reallocation algorithms by implementing the optimizeAllocation method of the PowerVmAllocationPolicyAbstract class, which is called at every time frame and passed with the full set of current VMs in the data center.

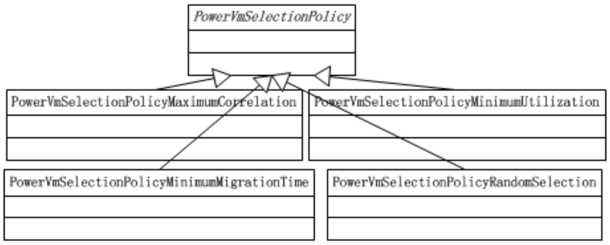

Figure 3. VM selection policy

C. VM Selection Policy

The VM selection problem consists of defining, among the VMs in a certain host, which one should migrate to a new machine requested by a user because of the power consideration. PowerVmSelectionPolicy is an abstract class that represents the VM selection policy. CloudSim 3.0 defines four VM (Figure 3) selection policies for the host to choose which VM should be migrated. The Minimum Migration Time (MMT) policy select a VM that requires the minimum time to complete a migration relatively to the other VMs allocated to the host. The Random Choice (RC) policy selects a VM to be migrated according to a uniformly distributed discrete random variable. The Maximum Correlation (MC) policy selects those VMs to be migrated that have the highest correlation of the CPU utilization with other VMs. The Minimum Utilization (MU) policy selects a VM to be migrated that requires the minimum CPU utilization relatively to the other VMs in the host.

To define your own VM selection policy, extend PowerVmSelectionPolicy to define the new selection policy, and pass this object in the initialization of Host.

D. Implement Your Own Algorithms

There are several places in CloudSim where you can implement your algorithm according to what the algorithm is intended to do. Except Scheduling policy, VM provisioning policy and VM selection policy, which describe above, here are several other classes that you may need to modify or extend in order to make your own simulation:

- DatacenterBroker – this class only submit a list of VMs to be created and schedules Cloudlets sequentially on them. Modifying the way VM provisioning requests are submitted to datacenters and the way cloudlets are submitted and assigned to VMs, researchers can create a new Broker class which extends DatacenterBroker class.
- Datacenter – this class behaves like an IaaS provider: it receives requests for VMs from brokers and creates the VMs in hosts.

To change the default behavior of DatacenterBroker and Datacenter, researchers can either extend these classes to add the intended behavior, or implement new ones from scratch. In the latter case, these new entities have to extend SimEntity and implement the processEvent (SimEvent) method.

- PowerVmAllocationPolicyMigrationAbstract – this class implement a power-aware dynamic VM consolidation algorithms that use VM live migration to dynamically reallocate VMs at every time frame. Researchers can implement new power-aware dynamic VM consolidation algorithms by extending this class and overriding the optimizeAllocation method.

V. Conclusion and Future Work

Using CloudSim to simulate a cloud computing datacenter avoids spending time and effort to configure a real testing environment. This paper introduced the CloudSim simulator including its architecture, and how to use it to model the cloud environment. In the future, we will use CloudSim platform to evaluate algorithm which aim at improve the average resource utilization of the cloud datacenter and reduce the energy consumption.

Acknowledgment

This work is supported by National Natural Science Foundation of China under grant No.61232009, which name is the basic theory and key technologies of the low power and secure embedded processor. And We like to acknowledge our thanks to R. Buyya, from Cloud Computing and Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia for providing good research articles on his portal related to cloud computing.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[2] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," Oakland, CA: Analytics Press. August 1.http://www.analyticspress.com/datacenters.html, Tech. Rep., 2011.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. Khan, "Greencloud: A packet-level simulator of energy-aware cloud computing data centers," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1–5.

[5] S.-H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. Das, "Mdcsim: A multi-tier data center simulation, platform," in *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on*, 2009, pp. 1–9.

[6] D. Meisner and T. F. Wenisch, “Stochastic queuing simulation for data center workloads,” *Proc. of the Workshop on Exascale Evaluation and Research Techniques (EXERT)*, 2010.

[7] R. Ltd, “Welcome to the dcsg simulator,” http://dcsg.bcs.org/welcome-dcsg-simulator.

[8] F. Julia, J. Roldan, R. Nou, J. O. Fito, A. Vaque, I. Goiri, and J. L. Berral, “Eefsim: Energy efficiency simulator,” Dept.of Computer Architecture, Tech. Rep., 2010.

[9] S. K. S. Gupta, R. Gilbert, A. Banerjee, Z. Abbasi, T. Mukherjee, and G. Varsamopoulos, “Gdcsim: A tool for analyzing green data center design and resource management techniques,” in *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011, pp. 1–8.

[10] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” in *Concurrency Computation Practice and Experience*, vol. 24, no. 13. Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley and Sons Ltd, 2012, pp. 1397–1420.

[11] “Specpower_ssj2008 dell inc. poweredge c6105,” http://www.spec.org/power_ssj2008/results/res2013q1/power_ssj2008-20121218-00591.html, 2 2013.

[12] “Specpower_ssj2008 dell inc. poweredge c6145,” http://www.spec.org/power_ssj2008/results/res2013q1/power_ssj2008-20121218-00592.html, 2 2013.