



INTRODUCTION TO CLOUDSIM

Dejan Bulaja,
Kristina Božić,
Nikola Penevski
Nebojša Bačanin Džakula

Singidunum University,
Belgrade, Serbia

Abstract:

Cloud computing is a subscription-based service where networked storage space and other computer resources can be obtained. Due to its high availability, easy accessibility, scalability and adaptability, cloud computing is highly desirable in the rapidly growing world of computer technology. Because of this high demand, the Cloud Computing and Distributed Systems (CLOUDS) Laboratory built the CloudSim framework, which became very popular open source cloud simulator among the researchers and students. CloudSim is new simulation framework which allows uninterrupted modeling and simulation of cloud computing infrastructure. It helped open up the possibility of evaluating the hypothesis in a controlled environment where experimental results can be reproduced easily. This basic goal of this paper is helping the researchers to understand CloudSim's most important functions and its practical usage, and to take insights on hands-on example of creating a simulated cloud environment.

Keywords:

CloudSim, Cloud Computing, Job Scheduling, Load Balancing.

1. INTRODUCTION

Cloud computing provides affordable, scalable and ubiquitous computing over distributed networks. It is based on reliable, secure, fault-tolerant, sustainable and scalable infrastructure for hosting Internet-based application services. Cloud computing often experiences problems of different load and energy performance, while modeling and scheduling different applications and services for the cloud infrastructure, which can be solved by using CloudSim. CloudSim is a self-contained platform that provides an extensible simulation toolkit which enables modeling and simulation of cloud computing systems and application provisioning environments. CloudSim has the following basic properties that make it convenient for simulating cloud:

1. Provides controlled and a repeatable environment to create and simulate cloud entities
2. Can be extended to include user defined policies for cloud

Correspondence:

Nebojša Bačanin Džakula

e-mail:

nbacanin@singidunum.ac.rs



3. Provides changeable infrastructure modeling, changeable network architecture, federated cloud support
4. Provides VM provisioning, Host provisioning, network provisioning and application provisioning.

2. BACKGROUND

Cloud Computing

Cloud computing can be defined as “a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resourced based on service-level agreements established through negotiation between the service provider and consumers” [1].

The National Institute of Standards and Technology (NIST) defines cloud computing as: a pay-per-use model for enabling available, convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [2].

Some examples of popular cloud computing infrastructures include Amazon Web Services, Microsoft Azure, Google Cloud Platform and Heroku [3].

Cloud Computing Models

Cloud Computing Models can be grouped into three categories listed below:

1. Software as a Service (SaaS) - this model offers a complete application to the customer. A single instance of the service is running on the cloud and multiple end users can be served simultaneously.
2. Platform as a Service (PaaS) - in this model, a layer of software or development environment is offered to end users as a service. Utility is that the customer can build his own application which runs on the provider's infrastructure freely.
3. Infrastructure as a Service (IaaS) - basic storage and computing capabilities as standardized services over the network are provided by this model. Servers, storage systems, data center space are made available to customers to handle workloads.

Enterprises can deploy application on one of the three clouds, and those are Public, Private and Hybrid clouds. Public clouds are owned and operated by third party. All customers share the same infrastructure pool with limited security protection and configuration [4], but the cost of this service is low and supports “Pay-as-you-go” model. Private clouds, on the other hand, are built exclusively for a single enterprise. The aim is to address the existing concern of data security and offer greater control over the infrastructure. Hybrid clouds are a combination of both previously addressed cloud models. By allowing workloads to move between private and public clouds as computing needs and costs change, hybrid cloud gives businesses greater flexibility and more data deployment options [5].

3. CLOUDSIM ARCHITECTURE

CloudSim provides an emulation of all the processes that happen in the cloud as shown in figure 1. CloudSim contains 12 packages that are used to create a simulation. The most relevant one is called cloudsim and contains the classes for modeling the various cloud entities like a Datacenter, a Host, a Cloudlet and VMs.

Datacenter is a virtual representation of a real datacenter and contains several computing nodes known as Hosts that are responsible for providing the services desired by the end user. It is described by the following parameters:

1. Total available processing power in MIPS (Million Instructions Per Second)
2. Total available RAM in MB (Random Access Memory)
3. Total storage represented in MB
4. Total bandwidth in mbps. Bandwidth is crucial for delivering burstable stable services [6].

Datacenter Broker represents a broker acting on behalf of a user. It hides VM management such, as vm creation, submission of cloudlets to this VMs and destruction of VMs.

Host executes actions related to management of virtual machines (e.g. creation and destruction). A host has a defined policy for provisioning memory and bw, as well as an allocation policy for Pe's to virtual machines. A host is allocated to a datacenter. It can host virtual machines.



Virtual machines are portable, inter-operable, logically independent. Each VM created is allocated to a Host where all its computational requirements are met. VM is described by the following properties:

1. CPU power represented in MIPS
2. Number of processing cores
3. Amount of RAM memory
4. Size of the virtual disk image
5. Bandwidth in mbps
6. Name of the virtual machine

Cloudlet is a representation of a submitted task in the cloud. Each cloudlet is assigned to a VM. Cloudlet has the following attributes:

1. Id of cloudlet that is used by the broker to assign them to a virtual machine
2. Length of a task that is to be executed represented in bytes
3. Output file size is the size of the cloudlet after executing.
4. Number of processors that are required to run the cloudlet

VmScheduler supports two ways of executing Cloudlets: time shared and space shared policies. In the time shared policy Cloudlets are run simultaneously while the MIPS of the VMs is divided evenly among them. In the space shared policy VMs will execute cloudlets in a chronological order using full capacity for each cloudlet.

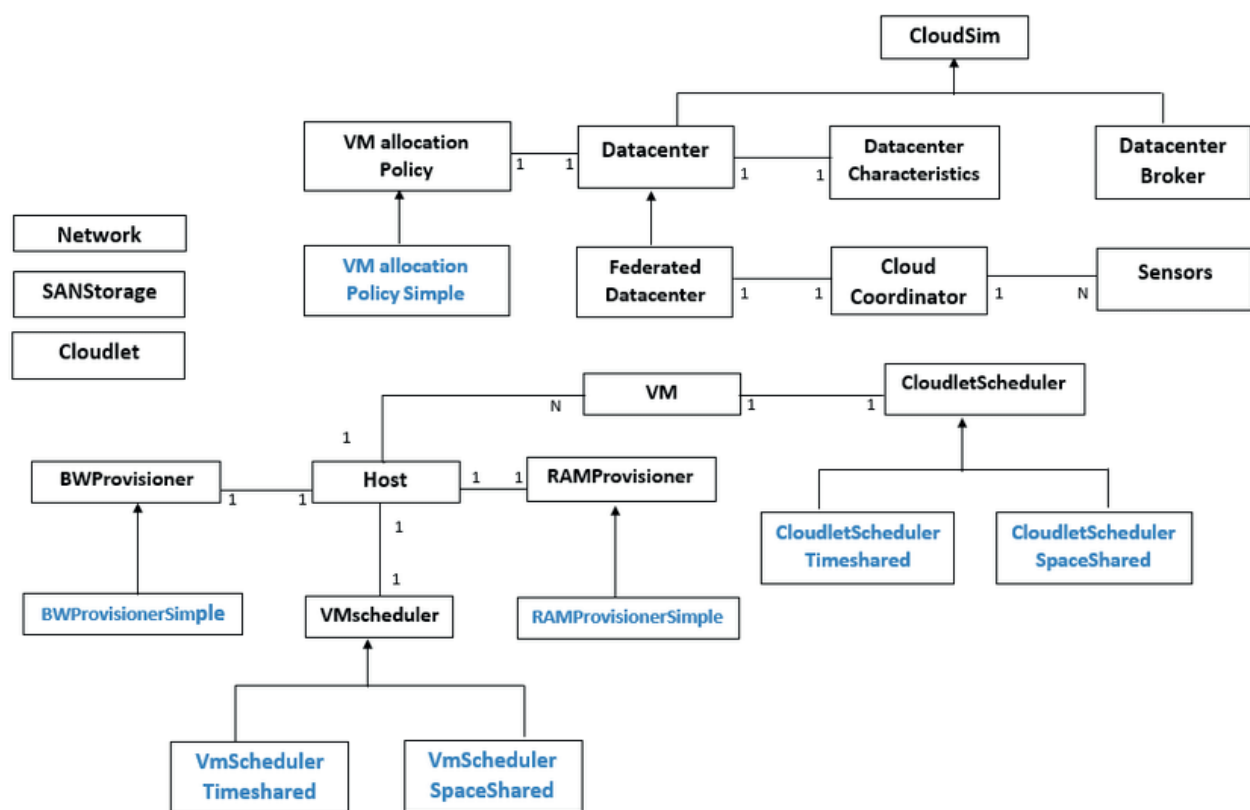


Figure 1. CloudSim Diagram

4. USING CLOUDSIM

CloudSim can be run as a standalone simulator to simulate cloud computing scenarios using any popular java IDEs. CloudSim can be used to research variety of existing policies, but it is also possible to change those policies for cloud operations, depending on the requirements.

Some of the most used Packages in CloudSim are listed below:

1. org.cloudbus.cloudsim.core - this package contains core functionalities which are required for simulation
2. org.cloudbus.cloudsim.distribution - contains variety of mathematical distributions, where Random Number Generator and Exponential Distribution are most common



3. org.cloudbus.cloudsim.network - where all the classes required for the network topology are contained
4. org.cloudbus.cloudsim.network, datacenter - which defines a network within a Datacenter
5. org.cloudbus.cloudsim.util - package contains classes that can provide specific mathematical functions
6. org.cloudbus.cloudsim.lists - containing the lists of operations on the list of resources like Hosts, VM, cloudlets and so on

All the packages and classes are contained within a jar (in folder CloudSim3.0.3\jars\CloudSim-3.0.3.jar). If one needs to add new policies to the existing toolkit, it must be done by overwriting the methods in existing classes. Some of the major classes along with their uses is shown in the figure 2. The class org.cloudbus.cloudsim.vAllocationPolicy is important because it can be extended and used for Host Load Balancing. Load Balancing is the process of spreading the load across available resources so as to decrease the overall response time of a task and to simultaneously increase the utilization of the resources [7].

Cloudlet	Host	Virtual Machine
getStatus()	allocatePesForVm()	getBw()
getCloudletId()	getAllocatedMipsForVm()	getHost()
getCloudletLength()	getId()	getId()
getCloudletOutputSize()	getDatacenter()	getMips()
getCostPerSec()	getBw()	getSize()
getProcessingCost()	getAvailableMips()	getCurrentAllocatedRam()
getVmId()	getNumberOfFreePes()	getCurrentAllocatedBw()
getWaitingTime()	getRam()	getCurrentAllocatedMips()
getUserId()	getStorage()	getCurrentAllocatedSize()
getExecStartTime()	getTotalMips()	getCloudletScheduler()
getFinishTime()	getVmList()	getVmm()
isFinished()	getBwProvisioner()	getUserId()
getCloudletFinishedSoFar()	getRamProvisioner()	getNumberOfPes()
getSubmissionTime()	addMigratingInVm()	isInMigration()

Figure 2. CloudSim Classes

Example of a CloudSim simulation

```

Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2
resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0

```

```

....
0.0: Broker: Trying to Create VM #19 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2,
Host #0
....
0.1: Broker: VM #19 has been created in Datacenter
#2, Host #1
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #2
....

```



```

0.1: Broker: Sending cloudlet 18 to VM #18
0.1: Broker: Sending cloudlet 19 to VM #19
0.1: Broker: Sending cloudlet 20 to VM #0
0.1: Broker: Sending cloudlet 21 to VM #1
.....
0.1: Broker: Sending cloudlet 37 to VM #17
0.1: Broker: Sending cloudlet 38 to VM #18
0.1: Broker: Sending cloudlet 39 to VM #19
2.0980547945205483: Broker: Cloudlet 0 received
2.239450143357758: Broker: Cloudlet 1 received
2.359450143357758: Broker: Cloudlet 2 received
....
5.098571022478637: Broker: Cloudlet 20 received
5.098571022478637: Broker: Cloudlet 21 received
5.098571022478637: Broker: All Cloudlets executed.
Finishing...
5.098571022478637: Broker: Destroying VM #0
.....
5.098571022478637: Broker: Destroying VM #19
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
    
```

```

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM  ID
Time  Start Time  Finish Time
0.1      0      SUCCESS      2    0    2
2.14     1      SUCCESS      2    1
.....
0.1      20     SUCCESS      2    0    5
0.1      21     SUCCESS      2    1    5
0.1      5.1
****Datacenter: Datacenter_0****
    
```

```

User id      Debt
4            21699.5
*****
    
```

Analysis of the simulation

In the given example a total of 40 Cloudlets are run across 20 VMs in a total of 2 data centers. The process starts by first creating the data centers and brokers for them. Brokers try to create the required amount of virtual machines in the data centers. In our example all VMs are created successfully in Datacenter_0 as it has enough resources. If some resources become insufficient due to the amount of VMs being created, e.g. not enough RAM for the creation of n-th VM, the broker will print an error that it's missing resources to create the given VM and try to create the next one. Even if not all VMs are created the simulation would continue with VMs that were successfully created. Broker will then assign each Cloudlet that is to be executed to a VM with a given scheduling algorithm. After receiving all the Cloudlets VMs would run simultaneously and simulate the time and resources that were needed to execute the Cloudlet. In the given example that start time is 0.1 ms and the finish time is 5.1 ms. It is worth noting that time in CloudSim is simulated and is not affected by real time and the speed that the program executes. It may take longer or shorter to finish but the time of finishing will not change when there are no random elements added to the equation.

Cost of simulation

Much like the pay-per-use model CloudSim supports simulation of the cost required to run tasks submitted by users [8]. Each data center has its attributes that describe the cost of running VMs. The cost is determined by the VM specifications and the time the VM ran in order to execute the given Cloudlets. Data center has costs per amount of memory, storage and processing power. These costs are multiplied by the VM specifications and the time the VM ran. In our example there are a total of 20 VMs so each of them is subjected to this calculation. The total cost or debt is the sum of costs of all VMs. It is displayed in an arbitrary unit. For the given example it is 21699,5. CloudSim is also capable of simulating different aspects multi-tenant data centers and calculating cost per user [9].



5. CONCLUSION AND FUTURE WORK

The progress of cloud technologies calls for novel approaches of better understanding processes that happen within the cloud infrastructure. CloudSim is a completely customizable tool that can simulate all parts of the cloud infrastructure making it suitable for testing different setups, algorithms, models and topologies in order to see the results, compare them and make improvements. After reading this work we hope that the reader has a firmer understanding of processes that happen on the cloud and ways to create and test their own.

Future works include incorporating a job scheduling algorithm such as cuckoo search algorithm (CSO) and particle swarm optimization (PSO) to achieve better load balancing of Cloudlets that are to be executed on VMs. The utilization of mentioned metaheuristics approaches instead of classical deterministic scheduling algorithms may significantly improve simulation results. With the current algorithms in CloudSim the way the Broker schedules Cloudlets is less than ideal. The time required to run multiple Cloudlets on multiple VMs can be decreased by using stochastic methods to find the near optimal order of running Cloudlets. Even the smallest improvements make a big difference on the large scale and produce large savings.

REFERENCES

- [1] R. Buyya, C.S. Yeo, S. Venugopal, J. Briberg, and I. Brandic, *Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility*, Future generation computer systems, Vol. 25, 2009, pp. 599-616.
- [2] *NIST Cloud Computing Reference Architecture*, 500-292, 2011. [Online]. Available: https://bigdatawg.nist.gov/_uploadfiles/M0008_v1_7256814129.pdf
- [3] Xiao-Li Wang, Ling-di Chen. *The Application of Cloud Computing in the Construction of Regional Digital Archives*, ARPN Journal of Systems and Software Vol. 5, Issue 2, 2015, pp 44-47.
- [4] C. Stergiou, K. E. Psannis, B. Kim, B. Gupta, *Secure integration of IoT and Cloud Computing*, Future Generation Computer Systems, Elsevier, Vol. 78, Part 3, 2018, pp. 964-975.
- [5] H. Talal, H. Noor, S. Zeadally, A. Alfazi, Q. Z. Sheng, *Mobile cloud computing: Challenges and future research directions*, Journal of Network and Computer Applications, Elsevier, Vol. 115, 2018, pp. 70-85.
- [6] Li, J. Cao, X. Wang, Y. Sun, Y. Sahni, *Enabling Software Defined Networking with QoS Guarantee for Cloud Applications*, 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, CA, 2017, pp. 130-137, doi: 10.1109/CLOUD.2017.25.
- [7] Suchintan Mishra, Manmath Narayan Sahoo. *On using CloudSim as a Cloud Simulator: The Manual*, 2017, doi: 10.13140/RG.2.2.30215.91041.
- [8] D. Alsmadi, V. Prybutok, *Sharing and storage behavior via cloud computing: Security and privacy in research and practice*, Computers in Human Behavior, Elsevier, Vol. 85, 2018, pp. 218-226.
- [9] K. Jeong, R. Figueiredo, K. Ichikawa, *PARES: Packet Rewriting on SDNEnabled Edge Switches for Network Virtualization in Multi-Tenant Cloud Data Centers*, 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, CA, 2017, pp. 9-17. doi: 10.1109/CLOUD.2017.11.