

Using CloudSim and DCSim for Hybrid Cloud Environment

Richa^{1*} Dr. Gagandeep²

¹ Research Scholar, Department of Computer Science, Punjabi University, Patiala, Punjab, India

² Professor, Department of Computer Science, Punjabi University, Patiala, Punjab, India

Abstract – Cloud Computing is an emerging IT environment with huge requirements of storage, resources and infrastructure. It enables cost effective resource usage and other business models. Computing today is shifting from hosting services in servers owned by individual organizations to data centres providing resources to a number of organizations on the shared platform. Managing such a data centre presents a unique set of goals and challenges. Through the use of virtualization, multiple users can run isolated virtual machines (VMs) on a single physical host, allowing for higher server utilization. In this we propose a hybrid environment using two different simulators. DCSim as private cloud and CloudSim as public cloud. In detail explanation of both simulators, how they are downloaded, installed and used.

Keywords: Cloud, Hybrid cloud, Data Centre, Simulator, Virtualization

-----X-----

I. INTRODUCTION

There is need for simulation tools to allow rapid development and evaluation of data centre management techniques. One of the biggest trends in computing today is the movement of processing from servers owned and operated by individual organizations to large-scale data centres. Virtualization is a technology that allows multiple Virtual Machines (VMs) to run on a single host system, providing the illusion that each has control of a physical machine. By packing many VMs onto a single host, its resources can be more utilized, thus saving on power, infrastructure and maintenance costs. Furthermore, a host's resources can be overcommitted, such that more resources are promised to each VM than are actually available. This can enable significantly higher utilization, but if one or more VMs exhibit an increase in their resource needs, it may lead to a situation where the host does not have enough resources to satisfy all of the demand. In that case, a VM must be migrated to a different host that has enough available resources to satisfy the VM's demand (Tighe, et. al., 2012).

Algorithms for dynamic resource management in the data centre have proven difficult to evaluate due to the scale and complexity of the infrastructure on which they are intended to run. As such, simulation is being accepted as a mean of rapidly evaluating new techniques at a speed and scale not possible with real implementations. One a technique has been

evaluated and fine-tuned using simulation, further experimentation can be performed using real infrastructure (Tighe, et. al., 2012).

CloudSim is a toolkit for simulating a data centre hosting a virtualized IaaS Cloud. Multiple users can create VMs within the data centre. It also provides the ability to simulate multiple data centres operating as a federation that are capable of coordinating resource allocation. CloudSim implements an HPC-style workload, with Cloudlets (jobs) submitted by users to VMs for processing. It can be used to simulate a transactional, continuous workload such as a web server or other service, but it lacks a detailed model of such an application (Tighe, et. al., 2012).

DCSim differs from other clouds in that it is focused on a virtualized data centre providing IaaS to any multiple tenants, similar to CloudSim. It differs from CloudSim in that it focuses on transactional, continuous workloads. As such, DCSim provides the additional capability of modelling replicated VMs sharing incoming workload as well as dependencies between VMs that are part of a multi-tiered application. SLA achievement can also be more directly and easily measured and available to management elements within the simulation (Tighe, et. al., 2012).

II. CLOUDSIM

CloudSim is the most efficient tool that can be used for modelling of Cloud. During the lifecycle of a Cloud, CloudSim allows VMs to be managed by data centres (Katyal & Mishra, 2013).

CloudSim provides architecture with four basic entities. These entities allow user to set-up a basic cloud computing environment and measure the effectiveness of Load Balancing algorithms. A typical Cloud modelled using CloudSim consists of following four entities Data centres, Hosts, Virtual Machines and Application as well as System Software. Data centres entity has the responsibility of providing Infrastructure level Services to the Cloud Users. They act as a home to several Host Entities or several instances hosts' entities aggregate to form a single data centre entity. Hosts in cloud are physical servers that have pre-configured processing capabilities. Host is responsible for providing software level service to the cloud users. Hosts have their own storage and memory. Processing capabilities of hosts is expressed in MIPS (million instructions per second). They act as a home to Virtual Machines or several instances of Virtual machine entity aggregate to form a Host entity. Virtual Machine allows development as well as deployment of custom application service models. They are mapped to a host that matches their critical characteristics like storage, processing, memory, software and availability requirements. Thus, similar instances of Virtual machine are mapped to same instance of a host based on its availability. Application and System software are executed on Virtual machine on-demand (Katyal & Mishra, 2013).

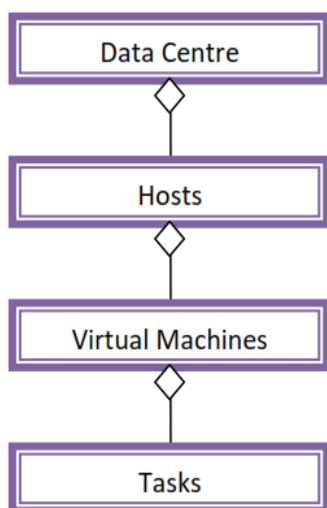


Fig. 1: Class Diagram of Cloud

Installation of CloudSim into Eclipse (Gagandeep, 2019)

- Download the CloudSim 3.0.3 from the website

<https://github.com/Cloudslab/cloudsim/releases>.

- Extract the files from downloaded zip file.
- Download 'Eclipse Java Photon' from <https://www.eclipse.org/photon/>.
- Install the Eclipse.
- When the installation process is finished, create a project with the name.
- Once you hit finish, an empty project with the mentioned name will be created in the package explorer
- Right click on the name there.
- A new window will open, now click on General > File System.
- Go to the directory where you have extracted your CloudSim.
- Select CloudSim and click finish.
- Now go to the link http://commons.apache.org/proper/commons-math/download_math.cgi.
- Download the file named commons-math3-3.6.1-bin.zip.
- Unzip this file after downloading.
- Now go to the left side of the eclipse tool in the project bar. Go to jar and right click on it. Click on import.
- Now go to the folder where you have placed the downloaded and extracted file. Then select all the jar files there and click finish.
- Run the simulator.

Simulation completed.

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	160	0.2	160.2
1	SUCCESS	3	1	160	0.2	160.2

Fig. 2 Creation of Data Centres, Cloudlets and Virtual Machines in CloudSim

III. DCSIM

DCSim is an extensible data centre simulator implemented in Java, designed to provide an easy framework for developing and experimenting with data centre management techniques and

algorithms. It is an event-driven simulator, simulating a data centre offering IaaS to multiple clients.

DCSim Metrics

The following section outlines some of the metrics being computed for each simulation (Keller et. al., 2013)

- ▶ **SLA violation:** When an VM requires more resources than is available on it, the VM will experience performance degradation. DCSim considers this to be an SLA (Service Level Agreement) violation and the percentage CPU resource required by running VMs but not available is recorded.
- ▶ **Data Centre Utilization:** The overall utilization of the data centre is calculated as the percentage of total CPU capacity in the data centre that is currently in use.
- ▶ **Active Hosts:** DCSim records the minimum, maximum and average number of hosts that are ON at any given time during simulation.
- ▶ **Host-hours:** Host-hours is the combined total of the active time of every host in the simulation. That is, if 10 hosts were active for 30 simulation minutes each, then 5 host-hours was used. This gives a combined measure of the number of hosts that were required to meet the workload demand throughout the entire simulation run.
- ▶ **Active Host Utilization:** DCSim measures the CPU utilization of all hosts that are currently in the ON state. The higher the average utilization, the more efficiently resources are being used.
- ▶ **Number of Migrations:** The number of VM migrations triggered during the simulation is recorded.
- ▶ **Power Consumption:** Power Consumption is calculated for each host, and the total kilowatt-hours consumed during the simulation are reported (Keller et. al., 2013).

DCSim generates a log containing details of the simulation as it progressed, including metric values at regular time intervals, host and VM states and other events. This log can be used to generate statistics, graphs or other visualizations in order to fully understand the behaviour of the simulated data centre and management algorithms. A visualization tool is included with DCSim which automatically generates a set of useful graphs based on the simulation log (Keller et. al., 2013).

DCSim Architecture

Fig. 3 outlines the main components of DCSim. The primary class is the Data centre, which contains Hosts, VMs and various management components and policies.

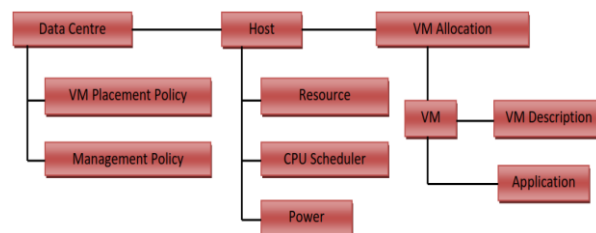


Fig. 3 Architecture of DCSim

Data Centre consists of a set of Host machines, which themselves contain a set of VMs. Data centre and host instances are governed by a set of policies and manager, which determine how they operate within simulation and provide extensible and customizable points to insert new management algorithms and techniques (Tighe, et. al., 2012).

In DCSim data centre consists of a set of interconnected physical host machines governed by a set of management policies. The purpose of data centre is to host a set of Virtual Machines (VMs) on its physical hosts, each with its own dynamically changing resource needs driven by an external workload (Tighe, et. al., 2012).

Each host maintains a collection of VM allocations, which represent a set of resources allocated to a single VM.

If a host's resources are being highly utilized, at or above some threshold, then the host is considered stressed. A stressed host is in an undesirable state, as any increase in resource requirements of a hosted VM could result in one or more VMs not being allocated enough resource to handle their workload (Tighe, et. al., 2012).

Each host may be in one of three primary states, ON, OFF or SUSPENDED, as well as traditional states between those three. A host in the ON state operates normally. When a host is OFF or SUSPENDED, any VMs it is hosting will not be allowed to execute. Furthermore, in the SUSPENDED state, a host consumes a small amount of power, and in the OFF state, a host consumes no power.

Each host contains a Power Model which defines how much power it consumes given its current CPU utilization.

On each host, its CPU, memory, bandwidth and storage are each managed by a separate Resource Manager. The resource managers' task

is to manage the allocation of resources to VMs. Resource managers can allocate a static amount of resource to VM, change the allocation of resources dynamically, as well as handle resource over committing, to suit particular management policies or goals. All four resource managers can be extended to suit the needs of a specific experiment.

While the CPU manager handles allocation of CPU, the CPU scheduler is responsible for actually scheduling the running of VMs on the host. There are several ways in which VMs can be scheduled to execute. The most basic method is to allow a VM to use at most its allocated number of CPU shares. This can lead to a waste of CPU resources, as any shares left unused by a VM cannot be used by another. The more common method is to allow other VMs to make use of these unused shares. This is referred to as a *work conserving* scheduler. DCSim currently implements a fair-share CPU scheduler, which gives each VM an equal opportunity to use CPU resources.

There is also a physical upper limit on the amount of CPU a VM can use. Each VM is created with a specific number of virtual cores, which the operating system of the VM interprets as real physical CPU cores. It is therefore not possible for a VM to consume more CPU than the full capacity of the number of virtual cores it contains (Tighe, et. al., 2012).

Installation of DCSim on Netbeans

- Download DCSim from the website <https://github.com/digs-uwo/dcsim>.
- After downloading, extract the files from downloaded zip file.
- Download 'Netbeans' from <https://netbeans.org/downloads/>. Netbeans IDE 8.2 has been used in this research, which is the latest version.
- Install the Netbeans.
- When the installation process is finished, create a project with the name.
- After that, add common maths 3-3.0 jar file in the project.
- The minimum requirement of java for this is JDK 1.8. Check in your computer for that. If it is not available or lower version is there, then download JDK 1.8 from <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.
- Add JDK 1.8 in the project library.

- Run the simulator.

```

run:
Data centre created.
DATA CENTRE: edu.uwo.csd.dcsim.DataCentre@452b3a41
CLUSTER: 1
HOST: 1
HOST: 2
HOST: 3
HOST: 4
HOST: 5
CLUSTER: 2
HOST: 6
HOST: 7
HOST: 8
HOST: 9
HOST: 10
CLUSTER: 3
HOST: 11
HOST: 12
HOST: 13
HOST: 14
HOST: 15
BUILD SUCCESSFUL (total time: 0 seconds)

```

Fig. 4: Available Data centres and hosts in DCSIM

VI. REFERENCES

- [1] M. Tighe, G. Keller, M. Bauer, H. Lutfiyya (2012). "DCSim: A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management," *6th International DMTF workshop on Systems and Virtualization Management (SVM / CNSM)*, pp. 385-392.
- [2] M. Katyal, A. Mishra (2013). "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment," *International Journal of Distributed and Cloud Computing*, pp.5-13, Vol. I.
- [3] G. Keller et. al. (2013). "DCSim: A Data Centre Simulation Tool," *IFIP/IEEE International Symposium on Integrated Network Management*, pp. 1090-1091.
- [4] Richa, Gagandeep (2019). "Load Balancing in Hybrid Cloud Computing," *Research Review International Journal of Multidisciplinary*, Vol. 4, pp.650-655.

Corresponding Author

Richa*

Research Scholar, Department of Computer Science, Punjabi University, Patiala, Punjab, India

richasharma743@gmail.com