**ASSIGNMENT NO.03**

**Aim:** Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

**Theory:**

**What is CloudSim?**

**CloudSim** is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modelling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

For example, if you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

**Benefits of Simulation over the Actual Deployment:**

Following are the benefits of CloudSim:

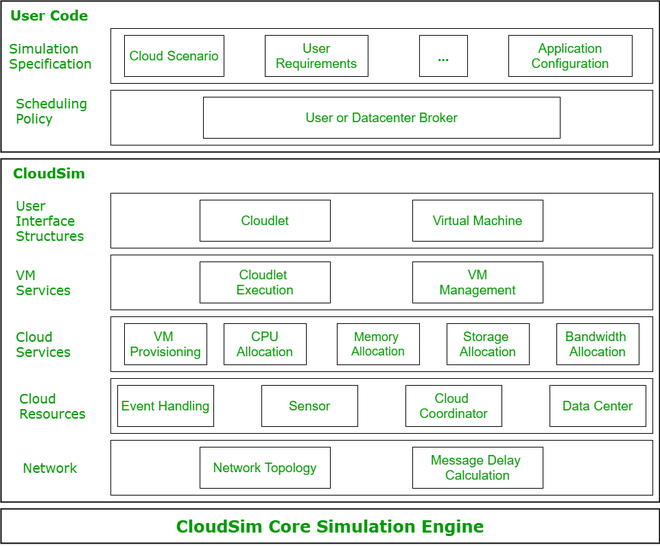
* **No capital investment involved**. With a simulation tool like CloudSim there is no installation or maintenance cost.
* **Easy to use and Scalable**. You can change the requirements such as adding or deleting resources by changing just a few lines of code.
* **Risks can be evaluated at an earlier stage**. In Cloud Computing utilization of real testbeds limits the experiments to the scale of the testbed and makes the reproduction of results an extremely difficult undertaking. With simulation, you can test your product against test cases and resolve issues before actual deployment without any limitations.
* **No need for try-and-error approaches**. Instead of relying on theoretical and imprecise evaluations which can lead to inefficient service performance and revenue generation, you can test your services in a repeatable and controlled environment free of cost with CloudSim.

**Why use CloudSim?**

Below are a few reasons to opt for CloudSim:

* *Open source* and *free of cost*, so it favours researchers/developers working in the field.
* Easy to download and set-up.
* It is more *generalized* and *extensible* to support modelling and experimentation.
* Does not require any high-specs computer to work on.
* Provides *pre-defined allocation policies* and *utilization models* for managing resources, and allows implementation of user-defined algorithms as well.
* The documentation provides *pre-coded examples*for new developers to get familiar with the basic classes and functions.
* Tackle bottlenecks before deployment to reduce risk, lower costs, increase performance, and raise revenue.

**CloudSim Architecture:**



*CloudSim Layered Architecture*

**CloudSim Core Simulation Engine** provides interfaces for the management of resources such as VM, memory and bandwidth of virtualized Datacenters.

**CloudSim** layer manages the creation and execution of core entities such as VMs, Cloudlets, Hosts etc. It also handles network-related execution along with the provisioning of resources and their execution and management.

**User Code** is the layer controlled by the user. The developer can write the requirements of the hardware specifications in this layer according to the scenario.

Some of the most common classes used during simulation are:

* **Datacenter**: used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
* **Host**: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
* **VM**: this class represents a virtual machine by providing data members defining a VM’s bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
* **Cloudlet**: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task’s execution time, status, cost and history.
* **DatacenterBroker**: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.
* **CloudSim**: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.

STEPS TO FOLLOW :

1. Download and install eclipse

2.Open eclipse and go to File -> new->java project-> give any project name-> next-> finish.

3.Expand project by clicking on project name (SJFCloudSim) arrow -> select src-> right click on src->new->package->give any name to package(Assignment3)->finish

4.Right click on Assignment3(package)->select show in -> system explorer->it will open src folder -> open Assignment3 package->download sample code from GitHub and paste 5 Files in Assignment3 package folder-> close

5.Go to eclipse -> right click on project name(SJFCloudSim)->select close project

6.Right click on project name(SJFCloudSim)->open project-> click on arrow before project name->click on src arrow->Assignment3 arrow->open all 5 files and write package name(Assignment3)in first statement of each file and save each file.

7.Select project tab on upper menu bar ->properties->from left side select java build path->select libraries -> click on add external JARs-> select already downloaded CloudSim JARs(CloudSim4.0 and CloudSim-examples4.0)->open ->apply->apply and close. If not downloaded then go to google ->CloudSim download-> select GitHub link releases –cloudslab/CloudSim->go to CloudSim 4.0-> select assets arrow-> download all 4 files -> extract CloudSim 4.0.tar

8.Run the file which contains main function(SJF\_Schedule.java) by right click -> run as -> java application.

OUTPUTS :

 <Images of actual implementation>

CONCLUSION :