CloudSim

Cloud Simulation Toolkit
Agenda

- Essentials to start with Cloudsim.
- Insight on Cloudsim modeled components.
- Insight on Cloudsim simulation process.
- Hands-on examples.
Introduction - Cloud Computing Service Stack
Cloud Infrastructure Challenges for researchers

- Cloud exhibit varying demands, supply patterns, system sizes and resources.
- Users have heterogeneous, dynamic and competing QoS requirements.
- Applications have varying performance, workload and dynamic application scaling requirements.
- due to third party ownership, no flexibility on configuration and cost.

re-producing reliable results and benchmarking is extremely difficult.
Cloudsim - a viable alternative

- Hasslefree extensible modeling and event based simulation of large scale cloud infrastructure with support of virtualization engine.
- Self contained platform for modeling:
  - Clouds,
  - Service brokers
  - Provisioning and allocation policies.
- Flexibility to switch between:
  - Space-shared
  - Time-shared allocation, at all the levels.
- Simulation of network connections among the simulated systems elements.
- Support for federated cloud environment.
Cloudsim - Essentials

- JDK 1.6 or above [http://tinyurl.com/JNU-JAVA](http://tinyurl.com/JNU-JAVA)
- Eclipse 4.2 or above [http://tinyurl.com/JNU-Eclipse](http://tinyurl.com/JNU-Eclipse)
- Alternatively NetBeans [https://netbeans.org/downloads](https://netbeans.org/downloads)
- Up & Running with cloudsim guide: [https://goo.gl/TPL7Zh](https://goo.gl/TPL7Zh)
Cloudsim-Directory structure

- cloudsim/  -- top level CloudSim directory
- docs/     -- CloudSim API Documentation
- examples/ -- CloudSim examples
- jars/     -- CloudSim jar archives
- sources/  -- CloudSim source code
Cloudsim - Layered Architecture

CloudSim core simulation engine

CloudSim architecture layers:
- User code
  - Simulation Specification
    - Cloud Scenario
  - User Requirements
  - Application Configuration

- Scheduling Policy
  - User or Data Center Broker

CloudSim components:
- User Interface Structures
  - VM Services
    - Cloudlet
    - Cloudlet Execution
  - VM Management
- Cloud Services
  - VM Provisioning
  - CPU Allocation
  - Memory Allocation
  - Storage Allocation
  - Bandwidth Allocation
- Cloud Resources
  - Events Handling
  - Sensor
  - Cloud Coordinator
  - Data Center
- Network
  - Network Topology
  - Message Delay Calculation
Cloudsim - Time/Space shared models

- - -

(a) VM-space, Task-space
(b) VM-space, Task-time
(c) VM-time, Task-space
(a) VM-time, Task-time
Cloudsim - Network latency matrix using BRITE

\[ E_{ij} = \text{Delay from entity}_i \text{ to entity}_j \]

\[
\begin{bmatrix}
0 & 40 & 120 & 80 & 200 \\
40 & 0 & 60 & 100 & 100 \\
120 & 60 & 0 & 90 & 40 \\
80 & 100 & 90 & 0 & 70 \\
200 & 100 & 40 & 70 & 0 \\
\end{bmatrix}
\]
Cloudsim - Network latency behavior

[Diagram showing the process of sending, making delay, and receiving messages in a network topology.]

- Send message
- Make delay
- Receive message

Message sender
Network Topology
CloudSim

Creating a message
Send a message
Receiving a message
Forward the message

do/ send a message
do/ calculate network delay
do/ put message in receiver's queue

Initial state
State
Event
Final state
Components of Virtualized Infrastructure
Cloudsim - Component model classes

- CloudInformationService.java
- Datacenter.java, Host.java, Pe.java
- Vm.java, Cloudlet.java
- DatacenterBroker.java
- Storage.java, HarddriveStorage.java, SanStorage.java
Cloudsim - Major blocks/Modules

- org.cloudbus.cloudsim
- org.cloudbus.cloudsim.core
- org.cloudbus.cloudsim.core.predicates
- org.cloudbus.cloudsim.core.distributions
- org.cloudbus.cloudsim.core.lists
- org.cloudbus.cloudsim.core.network
- org.cloudbus.cloudsim.core.network.datacenter
- org.cloudbus.cloudsim.core.power
- org.cloudbus.cloudsim.core.power.lists
- org.cloudbus.cloudsim.core.power.models
- org.cloudbus.cloudsim.core.provisioners
- org.cloudbus.cloudsim.core.util
Cloudsim - key components

- Datacenter
- DataCenterCharacteristics
- Host
- DatacenterBroker
- RamProvisioner
- BwProvisioner
- Storage
- Vm
- VMAllocationpolicy
- VmScheduler
- Cloudlet
- CloudletScheduler
- CloudInformationService
- CloudSim
- CloudSimTags
- SimEvent
- SimEntity
- CloudsimShutdown
- FutureQueue
- DefferedQueue
- Predicate and associative classes.
Ok... so how exactly this system works?
Simulation flow for basic scenario
Cloudsim - Core simulation framework
Task execution Queue management

![Diagram showing task execution queue management]

**DeferredQueue**
- Event1 – t1
- Event2 – t1
- Event4 – t2
- Event5 – t3
- Event6 – t3

**FutureQueue**
- Event4 – t2
- Event5 – t3
- Event6 – t3
- Event3 – t4

**Runtime**
- runClockTick()
Cloudsim - General steps to follow

- Initiate the cloudsim simulation.
- create a datacenter.
- create a datacenter broker.
- create VMs/cloudlet add it to respective lists.
- submit vm and cloudlet list to broker.
- start simulation.
- stop simulation.
- print the end results.
To Work on Cloudsim only thing you require is to know

---

Computers cannot think for themselves or assume anything. They can do exactly what you tell them to do.

So think like a programmer and model your work in Cloudsim
Cloudsim - Quick look inside
Cloudsim - Q & A

---

- @anupinder
- anupindersingh@superwits.com

- Share your feedback at: http://tinyurl.com/CloudSimJNUDec16