



# CloudSim

Muhammad Umar Hameed  
AIS Lab, KTH-SEECS

KTH  
Applied  
Information  
Security  
Lab



Department of Computing, School of Electrical  
Engineering and Computer Sciences, NUST -  
Islamabad



# Agenda

- Introduction
- Features of CloudSim
- Architecture of CloudSim
- SimJava
- GridSim
- Scheduling
- Cloudlets
- Latest Release
- Example Run

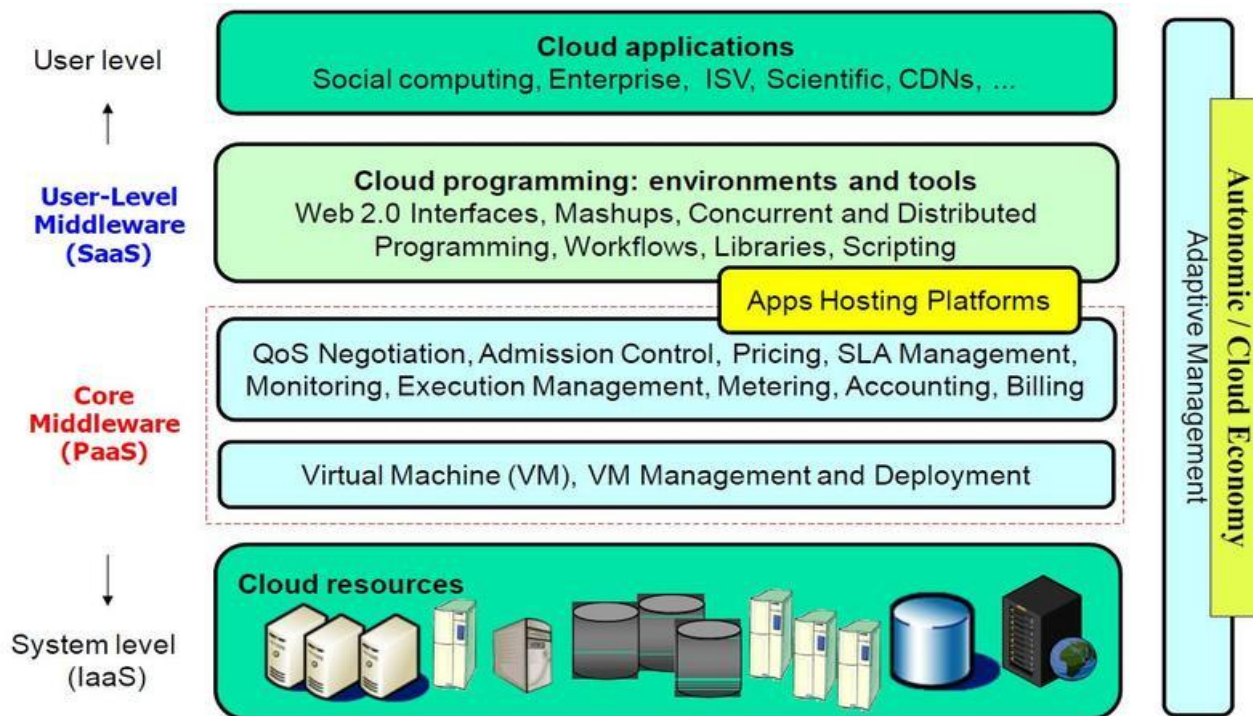
# INTRODUCTION

- Framework for simulation and experimentation Cloud infrastructure and application services
- Developed in CLOUDS lab, Computer Science lab, University of Melbourne.
- Latest Release: CloudSim Toolkit 3.0

# Features of CloudSim

- Modeling and Simulation of large scale Data Center.
- Energy-aware computational resources.
- Data Center Network Topologies
- User Defined Policies for allocation of hosts to virtual machines and policies for allocation of host resources to VM

# Layered Design of Cloud Computing



KTH  
Applied  
Information  
Security  
Lab



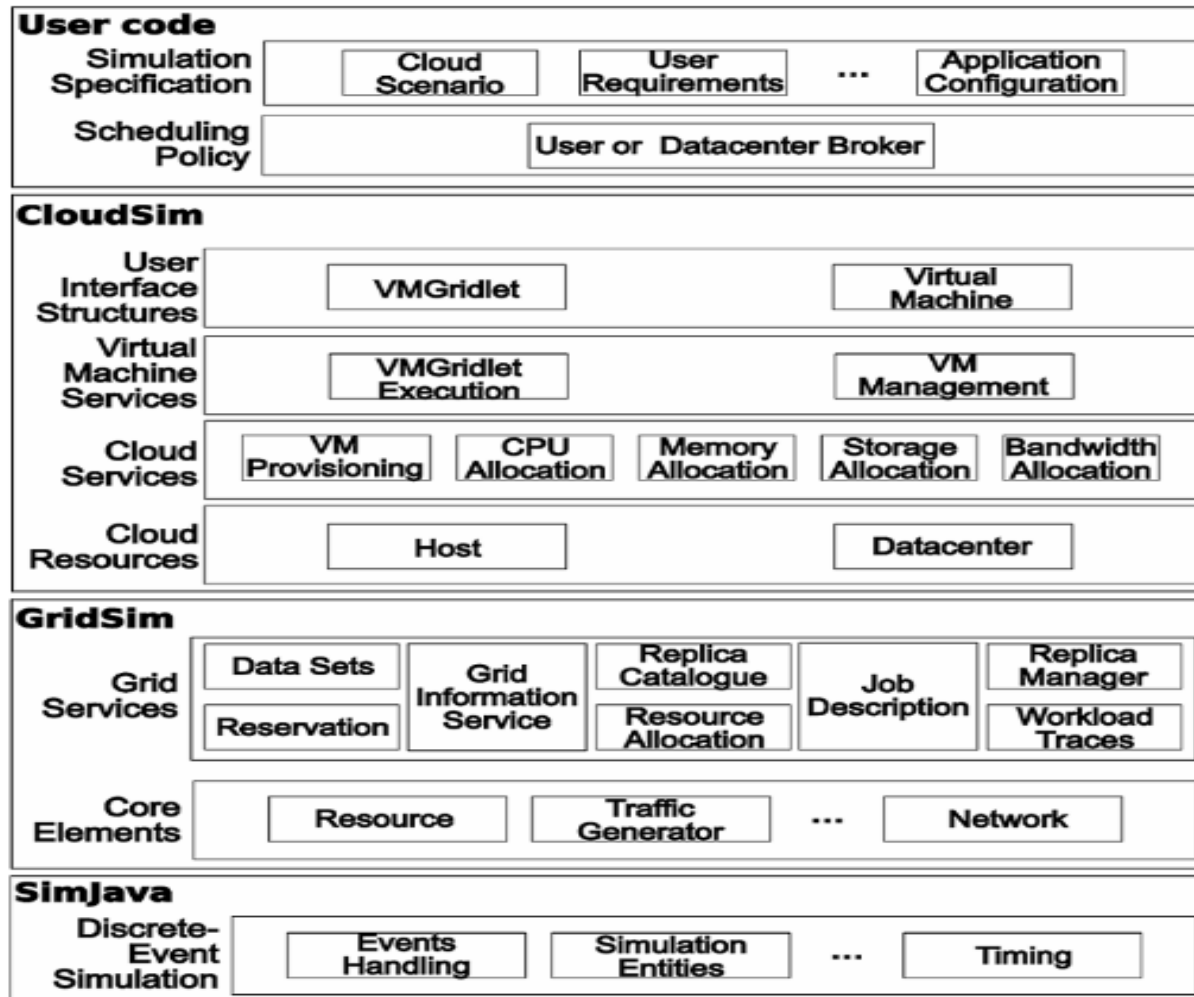
Department of Computing, School of Electrical  
Engineering and Computer Sciences, NUST -  
Islamabad



# Purpose of Cloudsim

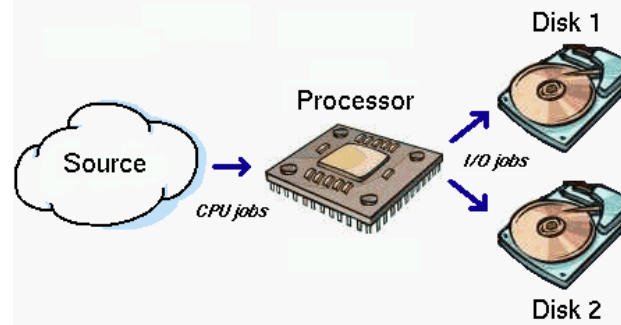
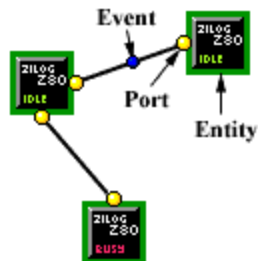
- The existing systems do not support mechanisms and policies for dynamically coordinating load among different data centers in order to determine the optimal location for hosting application services to achieve reasonable QoS levels.
- The CloudSim simulation layer provides support for modeling and simulation of virtualized Cloud-based data center environments including dedicated management interfaces for VMs memory, storage, and bandwidth.

# Architecture of CloudSim



# SimJava

- SimJava is a discrete event, process oriented simulation package.
- Each system is considered to be a set of interacting processes or entities as they are referred to in SimJava.
- These entities are connected together by ports and communicate with each other by passing events.





# GridSim

The GridSim toolkit supports modeling and simulation of a wide range of heterogeneous resources, such as single or multiprocessors, shared and distributed memory machines such as PCs, workstations and clusters with different capabilities and configurations. It can be used for modeling and simulation of application scheduling on various classes of parallel and distributed computing systems such as clusters.

# CloudSim

- The CloudSim is implemented at the next level by programmatically extending the core functionalities exposed by the GridSim layer.
- CloudSim provides novel support for modeling and simulation of virtualized Cloud-based data center environments such as dedicated management interfaces for VMs, memory, storage, and bandwidth.

# CloudSim: functionalities

- Transparently managing a cloud infrastructure such as provisioning of hosts to VMs based on user requests, managing application execution and dynamic monitoring handled
- Management interfaces for VMs, memory, storage and bandwidth.
- Manage instantiations and execution of core entities (VMs, hosts, datacenters and apps)

# Multi-layered design of the CloudSim software framework

- The top-most layer in the CloudSim stack is the User Code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their requirements), VMs, number of users and their application types, and broker scheduling policies.

# Various Modeling in CloudSim

- Modeling the cloud
- Modeling the VM allocation
- Modeling the cloud market
- Modeling the network behavior
- Modeling federation of clouds
- Modeling dynamic workloads
- Modeling data center power consumption
- Modeling dynamic entities creation

# Scheduling in CloudSim

- Host Level

VM scheduler

- VM Level

Cloudlet Scheduler

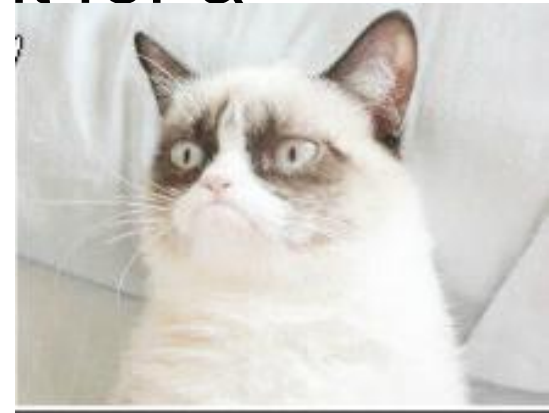
Sharing VM resources

- Broker Level(App scheduling)

Given a few Cloudlets from a broker

# What CloudSim Cannot do?

- Run Actual Applications for real.
- Run on its own.
- Can be a replacement for a cloud environment



# Cloudlets

A cloudlet is a new architectural element that arises from the convergence of mobile computing and cloud computing. It represents the middle tier of a 3-tier hierarchy: mobile device - cloudlet - cloud. A cloudlet can be viewed as a "data center in a box" whose goal is to "bring the cloud closer"



# Latest Release

1 - 12 of 12

	Filename ▼	Summary + Labels ▼	Uploaded ▼	ReleaseDate ▼	Size ▼	DownloadCount ▼	...
☆	<a href="#">cloudsim-3.0.3.tar.gz</a>	CloudSim 3.0.3: bug fix release <span>Featured</span>	May 2013	May 2013	9.9 MB	7490	
☆	<a href="#">cloudsim-3.0.3.zip</a>	CloudSim 3.0.3: bug fix release <span>Featured</span>	May 2013	May 2013	13.1 MB	13986	
☆	<a href="#">cloudsim-3.0.2.tar.gz</a>	CloudSim 3.0.2: bug fix release	Nov 2012	Nov 2012	9.9 MB	3845	
☆	<a href="#">cloudsim-3.0.2.zip</a>	CloudSim 3.0.2: bug fix release	Nov 2012	Nov 2012	13.0 MB	4117	
☆	<a href="#">cloudsim-3.0.1.tar.gz</a>	CloudSim 3.0.1: bug fix release	Oct 2012	Oct 2012	9.9 MB	631	
☆	<a href="#">cloudsim-3.0.1.zip</a>	CloudSim 3.0.1: bug fix release	Oct 2012	Oct 2012	13.0 MB	993	
☆	<a href="#">cloudsim-3.0.tar.gz</a>	CloudSim 3.0	Jan 2012	Jan 2012	9.9 MB	4104	
☆	<a href="#">cloudsim-3.0.zip</a>	CloudSim 3.0	Jan 2012	Jan 2012	13.0 MB	11870	
☆	<a href="#">cloudsim-2.1.1.tar.gz</a>	CloudSim 2.1.1: bug fix release	Feb 2011	Feb 2011	816 KB	1004	
☆	<a href="#">cloudsim-2.1.1.zip</a>	CloudSim 2.1.1: bug fix release	Feb 2011	Feb 2011	1.3 MB	3374	
☆	<a href="#">cloudsim-2.1.tar.gz</a>	CloudSim 2.1: new directory structure, code refactoring and clean-up, bug fixes	Jul 2010		819 KB	638	
☆	<a href="#">cloudsim-2.1.zip</a>	CloudSim 2.1: new directory structure, code refactoring and clean-up, bug fixes	Jul 2010		1.3 MB	2069	

# Example run CloudSim 3.0 in Eclipse

## Basic Examples

- [CloudSimExample1](#) – A simple example showing how to create a datacenter with one host and run one cloudlet on it.
- [CloudSimExample2](#) – A simple example showing how to create two datacenters with one host and a network topology each and run two cloudlets on them.
- [CloudSimExample3](#) – A simple example showing how to create two datacenters with one host each and run cloudlets of two users with network topology on them.
- [CloudSimExample4](#) – A simple example showing how to create two datacenters with one host each and run two cloudlets on them.
- [CloudSimExample5](#) – A simple example showing how to create two datacenters with one host each and run cloudlets of two users on them.
- [CloudSimExample6](#) – An example showing how to create scalable simulations.
- [CloudSimExample7](#) – An example showing how to pause and resume the simulation, and create simulation entities (a DatacenterBroker in this example) dynamically.
- [CloudSimExample8](#) – An example showing how to create simulation entities (a DatacenterBroker in this example) in run-time using a global manager entity (GlobalBroker).

## Network Examples

- [NetworkExample1](#) – A simple example showing how to create a datacenter with one host and a network topology and and run one cloudlet on it.
- [NetworkExample2](#) – A simple example showing how to create two datacenters with one host and a network topology each and run two cloudlets on them.
- [NetworkExample3](#) – A simple example showing how to create two datacenters with one host each and run cloudlets of two users with network topology on them.
- [NetworkExample4](#) – A simple example showing how to create a datacenter with one host and a network topology and and run one cloudlet on it. Here, instead of using a BRIE file describing the links, links are inserted in the code.

## Power Examples



Department of Computing, School of Electrical  
Engineering and Computer Sciences, NUST –  
Islamabad



# Related Projects

- CloudSimEx
- WorkflowSim
- SimpleWorkflow
- DynamicCloudSim
- RealCloudSim
- CloudReports
- CloudAuction
- CloudMIG Xpress
- CloudAnalyst

# Running Cloud Analyst in Netbeans

Configure Simulation

Define Internet Characteristics

Run Simulation

Exit

DC1

UB1

KTH  
Applied  
Information  
Security  
Lab



Department of Computing, School of Electrical  
Engineering and Computer Sciences, NUST -  
Islamabad





# Thank You

KTH  
Applied  
Information  
Security  
Lab



Department of Computing, School of Electrical  
Engineering and Computer Sciences, NUST -  
Islamabad

