Lab #4 – CloudSim

Introduction

Welcome to Lab #4.

Background

This lab introduces a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms - CloudSim.

Goals of Lab

- Create a data center and deploy a simple workflow management application in CloudSim

Pre-requisites

- Eclipse

Section 1 – Introduction to CloudSim

Why do we need simulation tool?

Due to the scale and complexity of shared resources, it is often hard to analyze the performance of new scheduling and provisioning algorithms on actual Cloud testbeds. Therefore, simulation tools are becoming more and more important in the evaluation of the Cloud computing model. Simulation tools allow researchers to rapidly evaluate the efficiency, performance and reliability of their new algorithms on a large heterogeneous Cloud infrastructure.

What is CloudSim?

The CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments.

What does CloudSim provide?

The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort.

Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios.

What is workflow?

In a cloud computing environment, applications and services can be decomposed into sets of smaller components, called jobs or tasks. The logical sequence of interdependent jobs (tasks) of an application forms a workflow.

A workflow is commonly represented by a Directed Acyclic Graph (DAG), denoted by $G = (V, E)$. Let the number of tasks in workflow be n. The set of nodes $V = \{T_1, \dots, T_n\}$ represents the tasks in the workflow applications, where n is the total number of tasks. The set of arcs $E = \{d_{ij}\}\ 1 \leq i, j \leq n$ represents the data dependencies among the tasks. An arc, $d_{ij} = (T_i, T_j) \in E$, implies that T_i transfers data to T_j. In this relationship, T_i is the parent task of T_j, and T_j is the child of T_i. The child task can be executed only after it receives data transferred from all of its parents. Fig. 1 shows a workflow example of 8 interdependent tasks. Note that any single task can have one or more children (except for the bottom nodes), and any single task can have one or more parents (except for the top node).

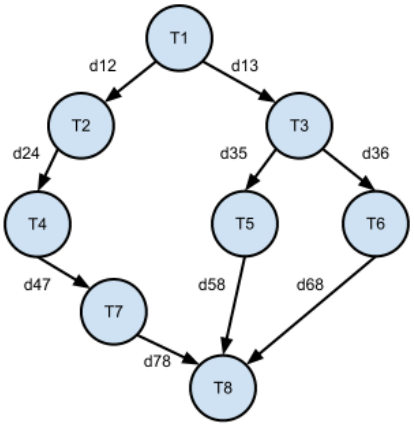


Fig. 1 A workflow example with 8 tasks.

What is workflow scheduling?

Workflow scheduling is one of the key components in a workflow management system. The scheduler decides which resources will be used, as well as which tasks will be executed on each of these resources. For example, consider a problem of 8 tasks and 5 resources (VMs). One possible mapping between tasks and resources is illustrated in Fig. 2.

T1	T2	T3	T4	T5	T6	T7	T8
R2	R4	R1	R5	R2	R3	R5	R1

Fig. 2 A mapping (8 tasks on 5 resources).

Section 2 – Download CloudSim documentation and source code

1. Go to website:

http://www.cloudbus.org/cloudsim/

Download the following two papers:

[1] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, *Software: Practice and Experience (SPE)*, Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011. (Preferred reference for CloudSim)

[2] Saurabh Kumar Garg and Rajkumar Buyya, NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations, *Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA)*, Melbourne, Australia, December 5-7, 2011.

2. Go to website:

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

   and download cloudsim-4.0 source code

   Note: cloudsim-4.0 requires Java 8

   Recommendation: Do this lab inside VM. No need to use Vagrant, just use VirtualBox to create and configure your VM instance of your favorite linux distro, and work inside it. You will have to download and install Java 8 and Eclipse within VM. Later, if some people need help with this (and with no help from classmates), I will post the instructions.

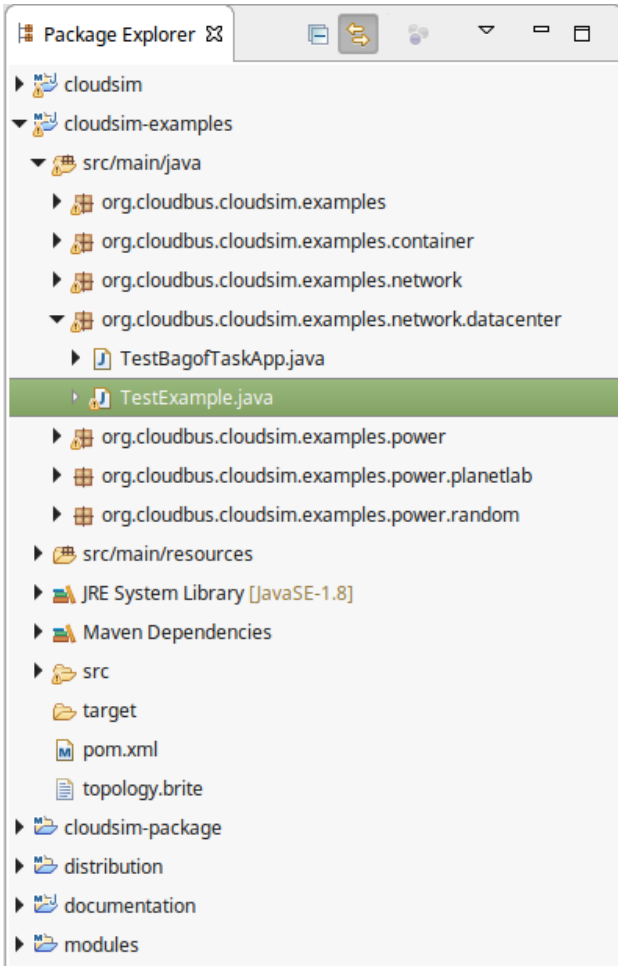
3. Create a java project and import the source code files into eclipse

Section 3 – Read through CloudSim documentation and Make sure you understand the following:

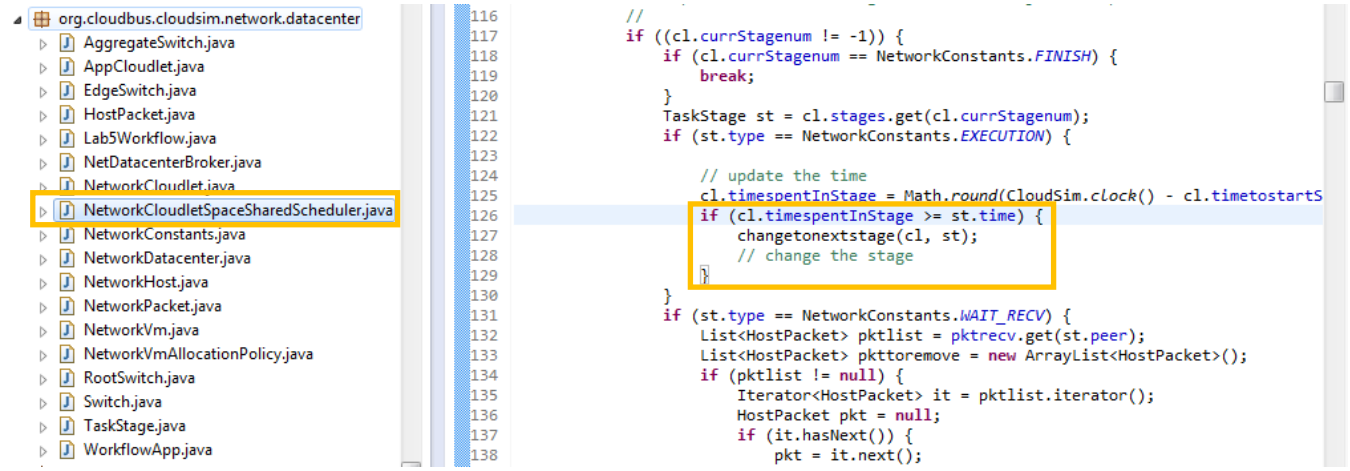
1. The CloudSim architecture and the NetworkCloudSim architecture
2. The Cloud modeling
3. The VMs and Tasks (Cloudlets) allocation
4. The Application modeling
5. The Network modeling (in NetworkCloudSim)
6. The function of each class in class diagram of CloudSim and NetworkCloudSim

Section 4 – Study and Run CloudSim source code

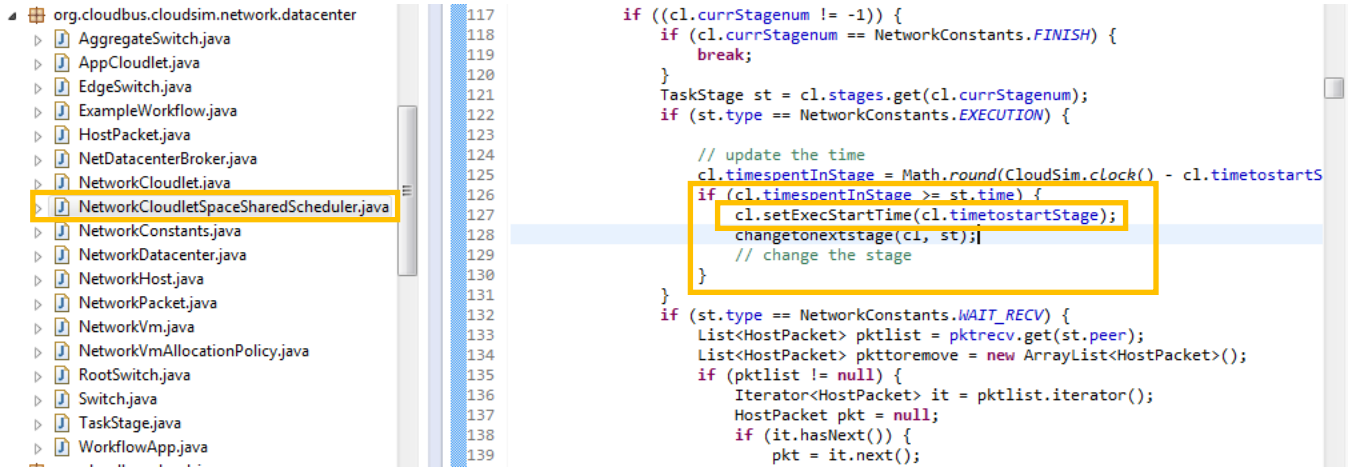
1. Run TestExample.java under examples/network/datacenter



2. Modify the code in NetworkCloudletSpaceSharedScheduler.java:



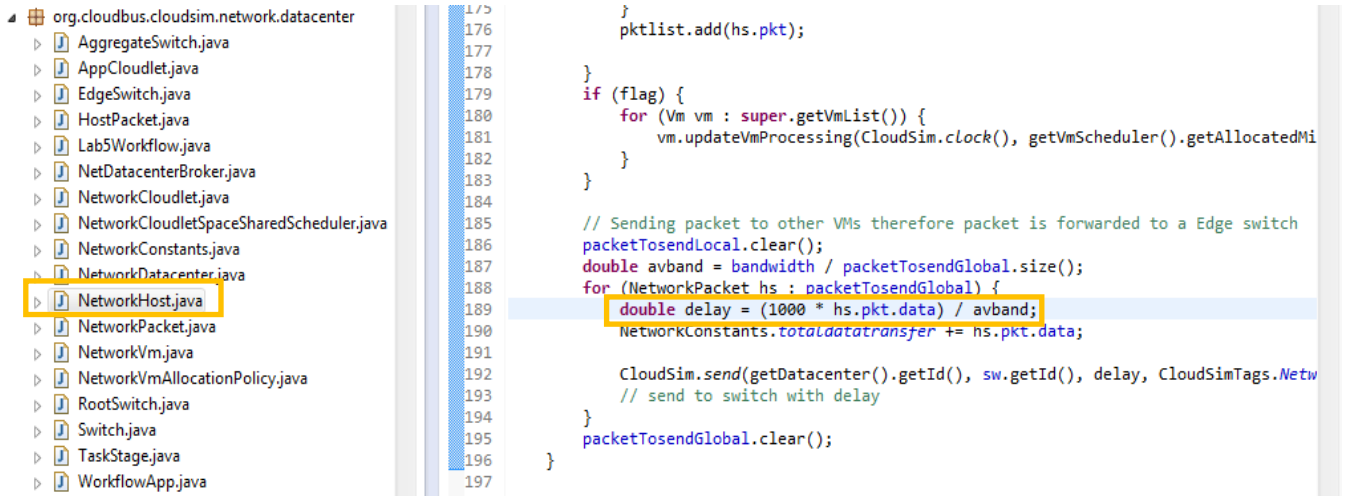
Insert a line “cl.setExecStartTime(cl.timetostartStage);” as following:



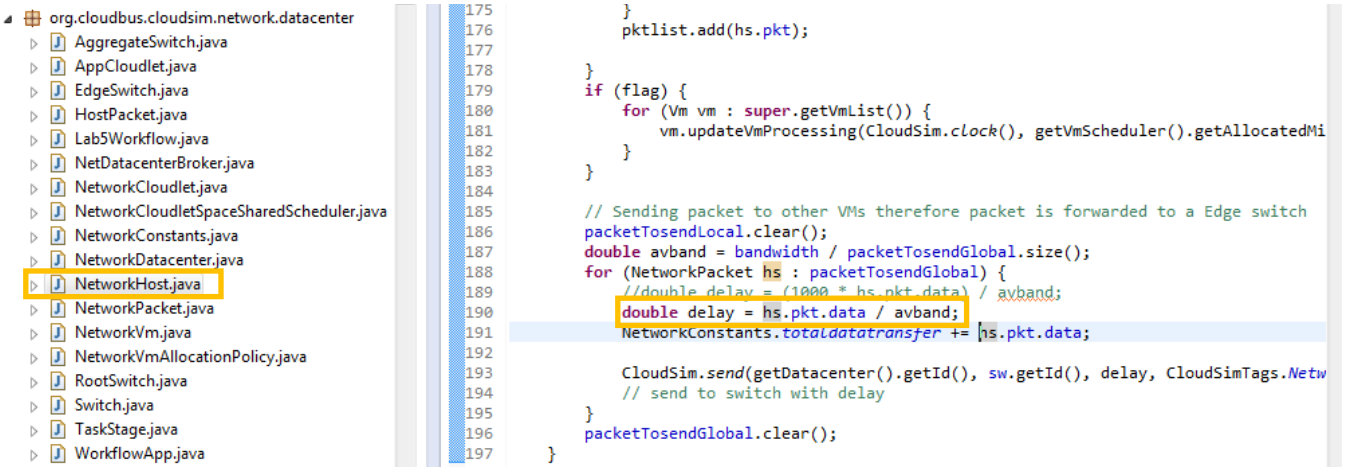
Run the TestExample.java again, you will see the difference of the start time of some cloudlets.

Note: This line is to set the start time of each cloudlet as its execution starting time.

3. Modify the code in NetworkHost.java:



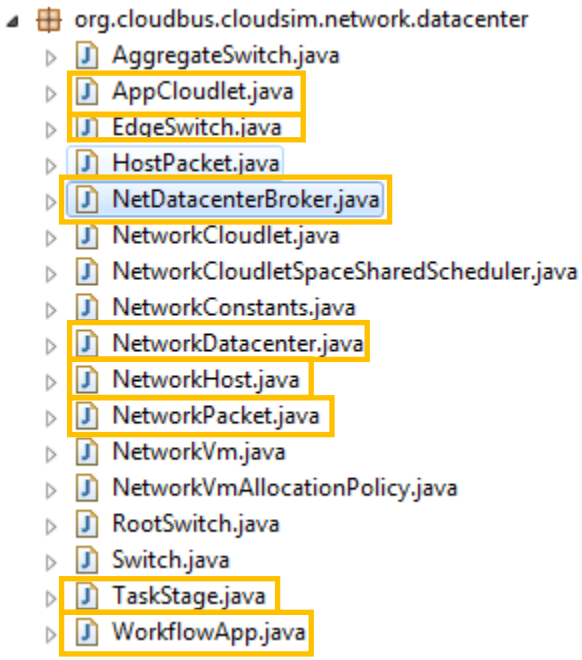
Change the line “double delay = (1000 * hs.pkt.data) / avband;” to “double delay = hs.pkt.data / avband;”



Note: This line is to calculate the transmission time from one host to the edge switch. To simplify the model, we don’t multiply the transmission time by 1000.

4. Make sure you understand the code in

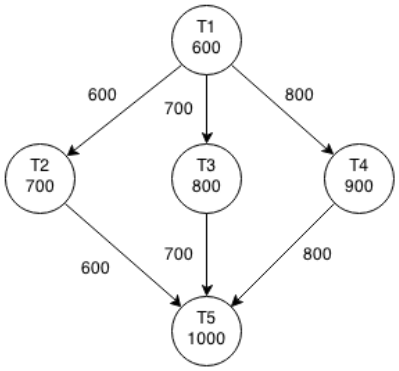
TestExample.java, NetDatacenterBroker.java, AppCloudlet.java, WorkflowApp.java, TaskStage.java, EdgeSwitch.java, NetworkHost.java, NetworkPacket.java, NetworkDatacenter.java



Homework

1. According to the example code in TestExample.java and other related files, answer the following questions:
   1) How many hosts and VMs are created in the data center? Draw the mapping between VMs and hosts?
   2) Draw the topology of the data center with bandwidth indicated?
   3) How many workflows are created? How many dependent tasks (cloudlets) are in each workflow?
   4) What is the duration for processing all the workflows?

2. Make change to workflow and workflow scheduling policy:
   1) Modify the code to simulate only one workflow, and the workflow is shown in the figure below:



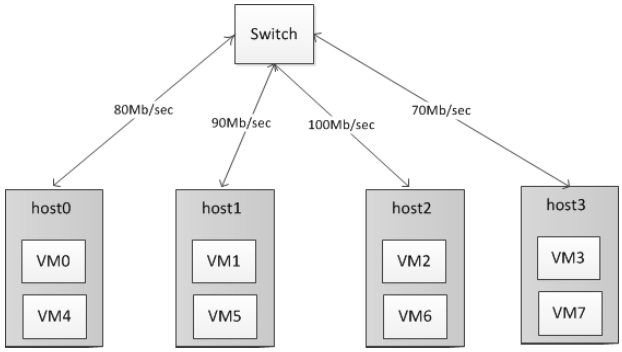
The workflow consists of five dependent tasks, each with computation in terms of millions of instructions, for example, task 1 (T1) has 600 million of instructions. Since the MIPS of VM is 1, we can calculate the execution time as 600 sec. There are data transmissions between tasks, for example, task 1 (T1) transfers 600MB data to task 2 (T2).

   2) Modify the code to make the mapping between tasks and VMs as following:

T1	T2	T3	T4	T5
VM3	VM4	VM0	VM2	VM6

   3) Keep other configuration unchanged, run the modified code, and paste the result.

3. (**Extra credit**) Make change to the network topology (hint: Method `CreateNetwork` in TestExample.java and Method processpacketforward in EdgeSwitch.java):
   1) Modify the code to create a network topology as following:



   2) Use the modified workflow and the mapping between tasks and VMs in the previous section, run the modified code, and paste the result.

Deliverables

- Create a {Microsoft Word| PDF } document containing the answers to the test section.
- Name the file <Last Name>_<First Name>_Lab04.{docx|pdf}
- Send the file to me via slack private message
  with the subject line “LAB## LastName FirstName” (ex.: LAB04 Smith John)