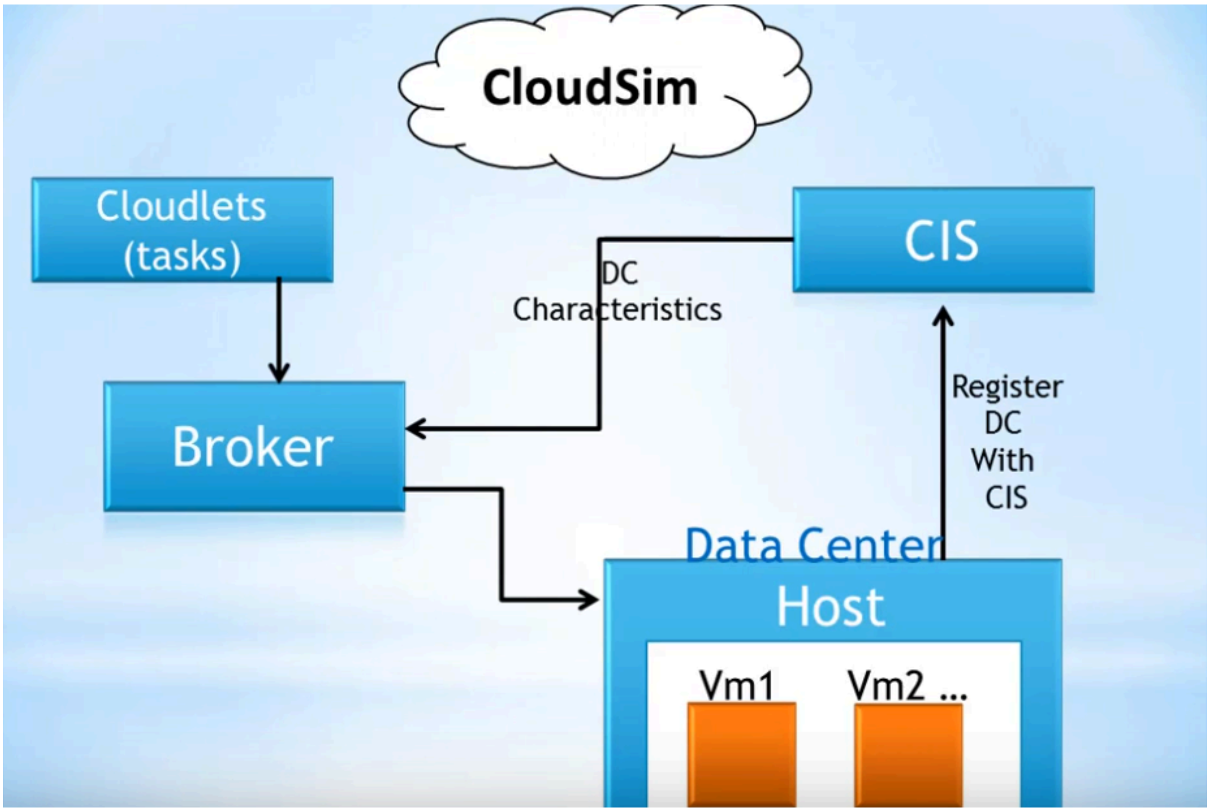
TP 2 - CloudSim review



Allocation scheduler policies

1. VM allocation policy

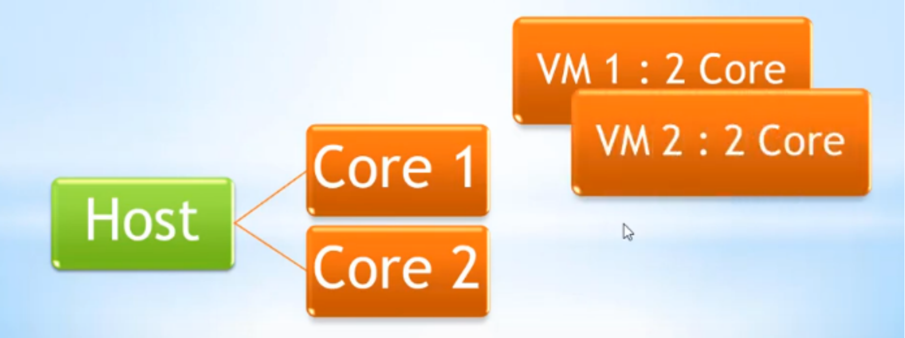
- Each *Vm* needs to be allocated to a *Host*, according to certain *algorithms* provided by the developer

1.1. VM allocation policy Simple

- Processor Element = Core
- The PE in *Host 1* for example is busy, so we allocate it to *Host 2*

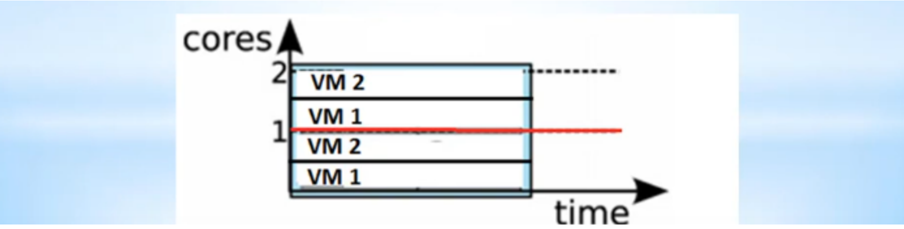
2. VM scheduler policy (*HOST*)

- How to allocate *Host* resources between *Vms*

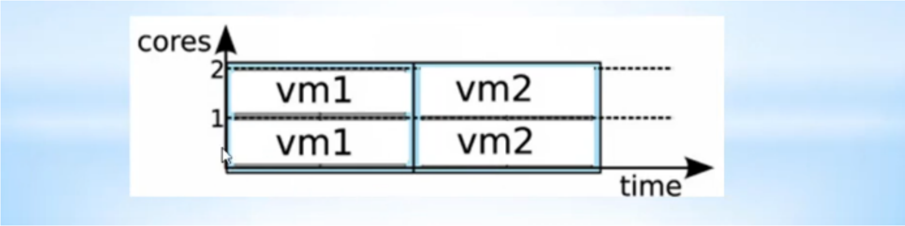


2.1. Time shared

- No queue

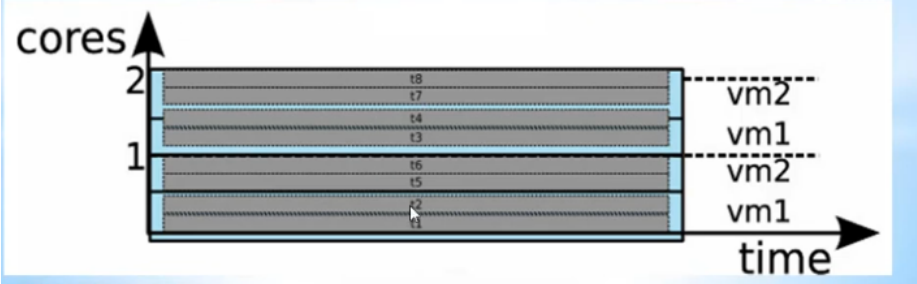


2.2. Space shared

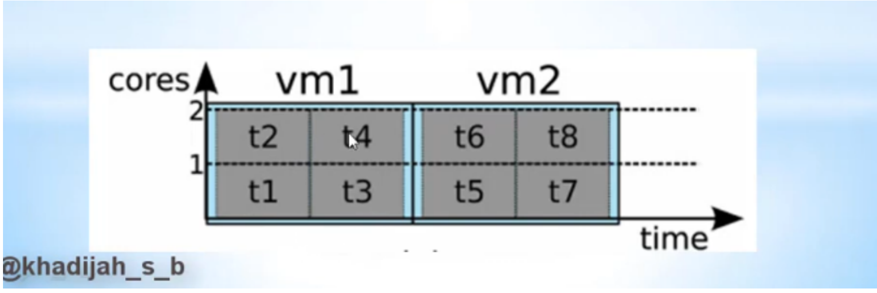


3. Cloudlet scheduler policy (*VM*)

3.1. Time shared



3.2. Space shared

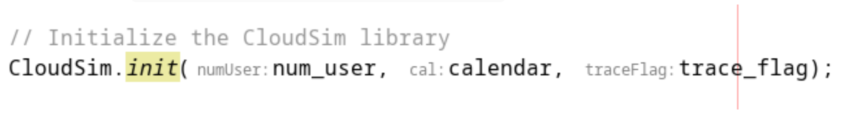


Simulation *12* steps

Step	Item	Parameters
1	Num. of User	Brokers
2	Common Variables	Time, Users
3	Object CIS	
4	DC instance	Host instance & Characteristics (PE cores, RAM, BW)
5	Broker instance	
6	Vm & Chars (PE,RAM,BW)	
7	Send VM info -> Broker	
8	Cloudlets (Tasks)	

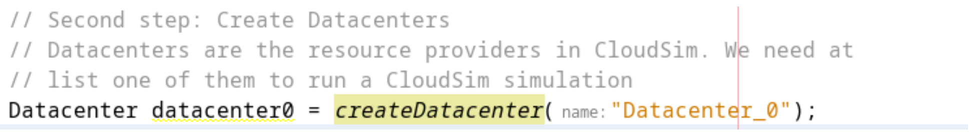
Step	Item	Parameters
9	Send Cloudlet -> Broker	
10	Start Simulation	
11	Stop Simulation	
12	Print Simulation	

3. Object CIS `CloudSimpleExample1.java`

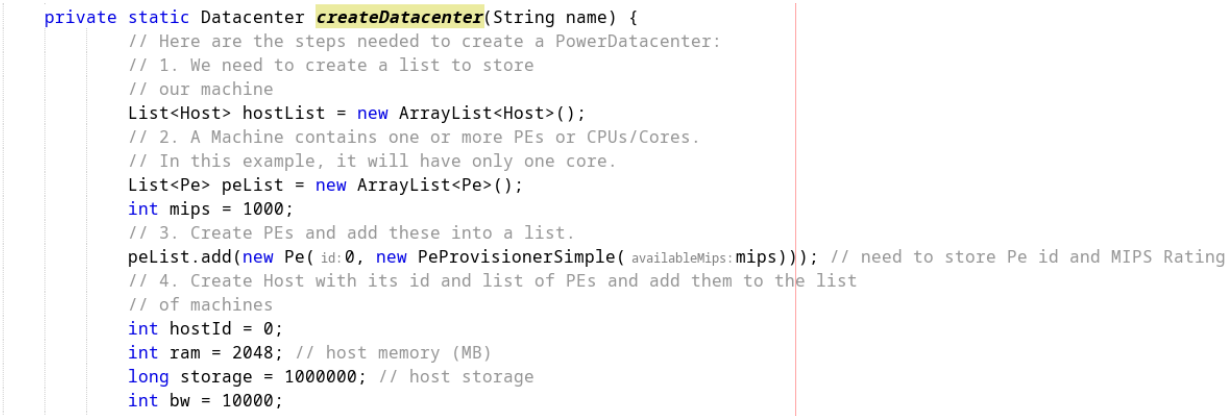


```
// Initialize the CloudSim library
CloudSim.init( numUser: num_user,  cal: calendar,  traceFlag: trace_flag);
```

4. DC Instance



```
// Second step: Create Datacenters
// Datacenters are the resource providers in CloudSim. We need at
// list one of them to run a CloudSim simulation
Datacenter datacenter0 = createDatacenter( name: "Datacenter_0");
```

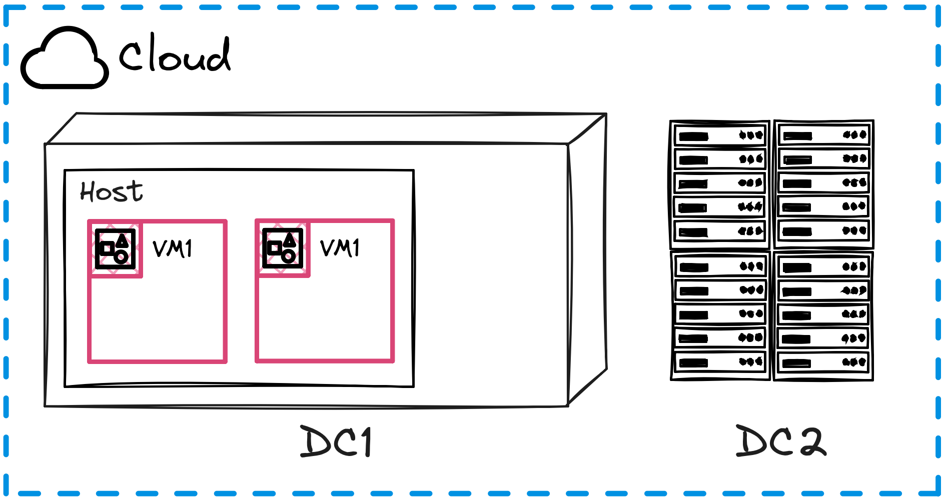


```
private static Datacenter createDatacenter(String name) {
        // Here are the steps needed to create a PowerDatacenter:
        // 1. We need to create a list to store
        // our machine
        List<Host> hostList = new ArrayList<Host>();
        // 2. A Machine contains one or more PEs or CPUs/Cores.
        // In this example, it will have only one core.
        List<Pe> peList = new ArrayList<Pe>();
        int mips = 1000;
        // 3. Create PEs and add these into a list.
        peList.add(new Pe( id: 0, new PeProvisionerSimple( availableMips: mips))); // need to store Pe id and MIPS Rating
        // 4. Create Host with its id and list of PEs and add them to the list
        // of machines
        int hostId = 0;
        int ram = 2048; // host memory (MB)
        long storage = 1000000; // host storage
        int bw = 10000;
```

II. Travail demandé :

□ A. Intégration du CloudSim dans NetBeans

□ Paramétrage de la simulation *(préparation de l'infrastructure du Cloud)*



- Paramètres de notre Cloud
  - N. de Datacenter
  - N. d'hôtes
  - Caractéristiques de chaque hôte (*Bw,CPU,RAM*)
  - Caractéristiques de chaque VM (*Bw,CPU,RAM*)

□ Simulation d’une application distribuée (application Client/Serveur)

- Application qui est composée de 2 parties
- Chaque partie doit s'exécuter sur une VM
- Puis les données sont échangées entre ces deux parties aussi entre les machines
- Il faut écrire une classe qui simule le comportement d’une application.
- Le client envoie des paquets ICMP vers le serveur et le serveur lui répond