CIS: Cloud Information System which is kind of a registry which contains the resources available on the cloud.

Resources of cloud:

Datacenter: each data center has some host and each host may have some set of virtual machines

Datacenter is required to be registered when it is created. This registration process is done by CIS

In our example our data centre has 1 host. That host has some hardware configurations:

No of processing elements (pe). PE may be 1,2,3 etc.

RAM, for e.g., 1gb, 2gb, etc.

Bandwidth

This cloud environment works on virtualization which differs from other technologies like distributed system, parallel computing etc.

This virtualization specifically says that this host will be virtualized in number of virtual machines. These virtual machines are inside this host. These virtual machines also have those hardware configurations (PE, RAM, BW).

After the data centre is registered by the CSI, there must be a broker which will summit tasks to the data centre.

This broker is basically a datacentre broker class which is responsible to summiting task to that particular data centre.

This broker is basically an entity which at the initial state talks to CIS and retrieve the resource information which are registered with this.

Datacentre is registered in the CIS.

CIS is passing datacentre characteristics to the Broker.

Once data centre broker has the datacentre characteristics and it has some tasks (which is called ‘cloudlets’ in cloudsim environment) (Cloudlets are submitted to the broker)

broker directly interacts with the data centre and assigns cloudlet(s) to some vms which are running in the hosts.

Datacentre, CIS and Broker are called entities in cloudsim environment.

More than one task (cloudlet) can be summitted to a virtual machine. The overall processing to be done, is performed in the physical machine.

The resources of the host are divided among the vms on the basis of ‘VM Schedular policies’. There are three policies in the cloudsim:

1. VM Allocation policy (This is used by data centre)
2. VM Schedular policy (This will be used by Host whenever processing is to be done by virtual machine.)
3. Cloudlet Schedular policy (This will be used by VM for the processing of the cloudlets)

All these policies are either time shared or space shared.

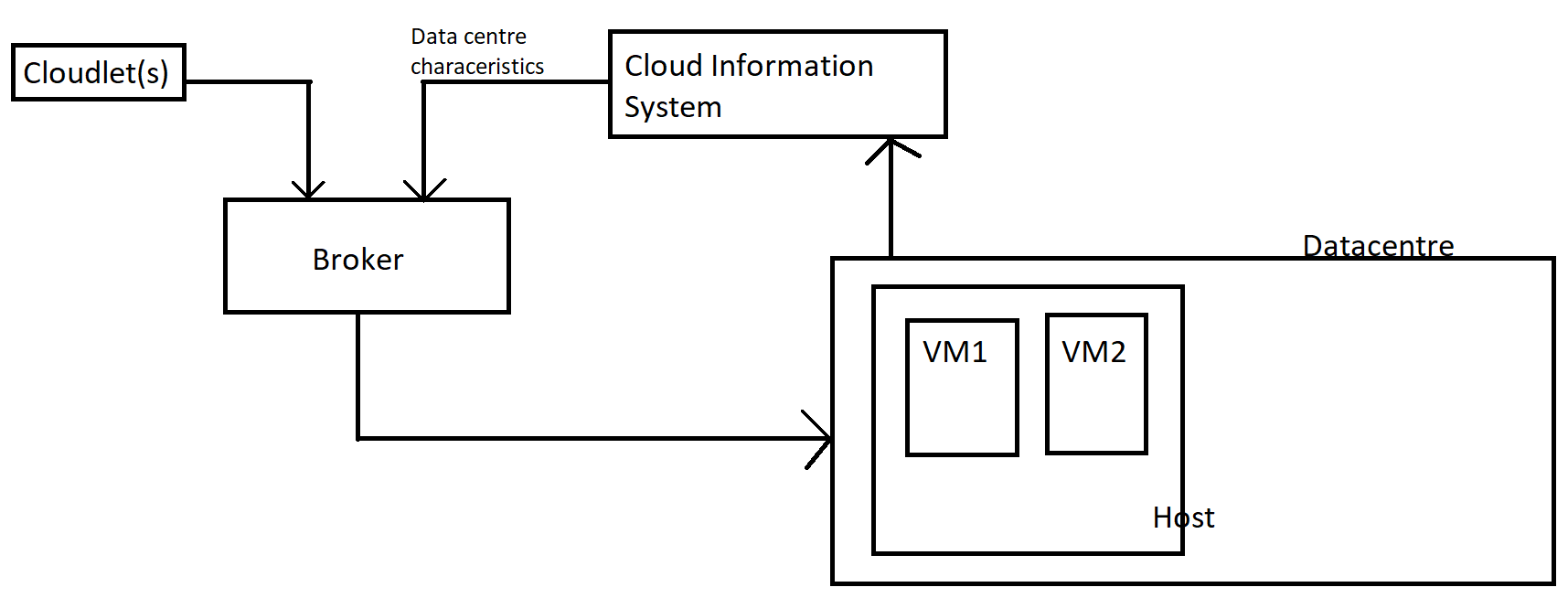
In time sharing policy a specific time will be allocated to each of the entity and in space sharing policy one by one VMs will be allocated.

There are also few more things like:

How the bandwidth will be provisioned.

How processing elements will be provisioned etc.

These are already available in cloudsim.



Available packages in Cloudsim:

Simulation components in org.cloudbus.cloudsim: Datacenter.java, DatacenterBroker.java, Cloudlet.java, Host.java, File.java, Storage.java etc.

Extending the Storage class HarddriveStorage.java, SanStorage.java etc.

Similarly Cloudlet is extended to ResCloudlet (it maintains all the objects and their operations which are submitted to some cloud resource for execution purpose) etc.

From the point of view of scheduling and utilization policy.

For the purpose of allocation VMs to host there are a pair of classes VmAllocationPolicy and VmAllocationPolicySimple. VmAllocationPolicy is an abstract class which is further implemented in VmAllocationPolicySimple class. VmAllocationPolicy class contains all those behaviours which may be required by Virtual Machine Manager (VMM) of a data center to allocate a VM to particular Host.

Similarly, VmScheduler is an abstract class which is implemented in VmSchedulerTimeShared and VmSchedulerTimeSharedOverSubscription class.

Also, similarly, UtilizationModel is an interface which is implemented in UtilizationModelFull, UtilizationModelNull, UtilizationModelPlanetLabInMemory.

CloudletSchedular is an abstract class which defines how the cloudlets will be executed in the VMs. Those are implemented in CloudletSchedularDynamicWorkload, CloudletSchedularSpaceShared, CloudletSchedularTimeShared

Simulation components in org.cloudbus.cloudsim.core:

This is the brain of these simulation process.

CloudSimTags: This particular class specifies about the event identifiers which are responsible for specific behaviours during this simulation. Here some static variables are defined which are very important as they go as a part of message and conveys the particular entity that this particular process is to be done.

DeferredQueue and FutureQueue:

These queue hold the events which are to be processed during the simulation