**Network Simulation**

**Architecture design:**

VM 4

VM 3

VM 1

VM 2

2

Data center 2

VM 3

VM 4

VM 1

VM 2

2

Data center 1

Root Switch connection

Implementation:

Initial plan was to implement network architecture with load balancing but network simulation framework in Cloudsimplus has limited functionality. Hence, we have implemented network simulation and load balancing simulation separately.

Since one of the limitations of network simulation is that the cloudlet (when active/executing) should have a VM associated with it the design choice we made is to use space shared policy for allocating VM to Host. If we allocate VM in timeshared way a cloudlet A sending data may send it to another cloudlet B which may or may not be running based on whether B’s VM has an allocated host or not, if B’s VM doesn’t have host then send activity fails and B keeps on waiting for data when it’s VM gets a host machine allocated (since data from cloudlet A never arrives). Hence VM migration or dynamic VM allocation is not feasible.

At physical level we implemented TOR topology, each host is connected to an Edgeswitch ,by default edge switch has 4 ports. We tried to use Aggregate switch and Root switch, as per documentation aggregate switch is used to connect Edgeswitch to datacenter/root switch, but there is no method or functionality to form that connection.

Application/Simulation consists of multiple cloudlets with each having execution task. Application simulates diffusing computation pattern with every 6th cloudlet sending data to 5 different cloudlets. In order to generate statistics, we compared the host bandwidth vs execution time, but found that Cloudsim is limited in the sense that it doesn’t consider host bandwidth while simulating the data transfer between different cloudlets. Number of packets sent is correlated with the execution time of the cloudlet as expected. Cloudlets are allocated in timeshared fashion when they are created. Since cloudlet needs to have VM associated with it for send or receive data task, each cloudlet is assigned to a VM uniformly.

References:

NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations (Saurabh Kumar Garg, Rajkumar Buyya)

<https://ieeexplore.ieee.org/document/6123487>

**Load Balancer Simulations**

**Overview:**

Created 5 Load Balancer simulations (FIFO/RoundRobin, Hashing, Random, MinMin, MaxMin). Ran all simulations and compared them based on time to tell which is the most efficient. Params for VM generation, Host generation and seed for dynamic Cloudlets generation are fetched from the configuration .conf file. The dynamically generated cloudlets are fed to different LB simulations.

**Design Choices:**

* While creating cloudlets I decided to make them SpaceShared (FullUtilizationModel), so that it will utilize all resource required and VM will not switch the executing cloudlet. This will also facilitate better comparison for different LB’s
* For creating dynamic cloudlets, I used UniformDistribution where seed is fetched from the params. For for the same seed the same cloudlets are generated which are used by different simulations. This is helpful as all simulations will be compared against the same dynamically generated cloudlet.

**Problems:**

I tried to code Oktopus algorithm first but there was difficulty in incorporating with network topology (simply there are constraints with the Cloudsim framework) such as:

* When creating network topology the network cloudlet requires a VM to be associated with it. This becomes a problem as all cloudlets become bound to a VM and VMMapper abstract ignores cloudlets if a VM is associated to them.
* Also tried to create a custom load balances which takes in account the next free cpu to assign cloudlet. The problem I faced was that in the cloudsim is that it doesn’t actually starts simulating when the cloudlets arrive (it starts after all the cloudkets are allocated). That is getExecVM method inside DataCenterBrokerSimple, DataCenterBrokerHeuristic, DataCenterBrokerAbstract classes gives initial VM state even if multiple cloudlets are allocated to it.

Results

