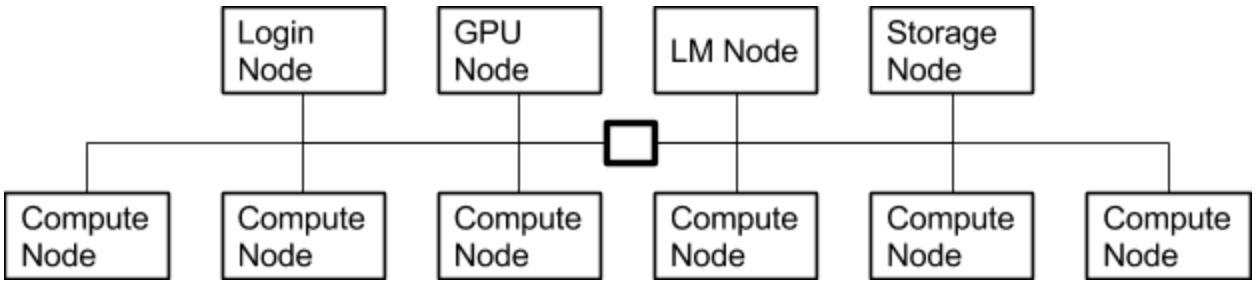
Project Assignment Part 02
Team 5-SDSC Comet
Luke Morrow, Grant Adams, Foster McLane, Thomas Rea

Diagram:



Network Topology Reasoning:

In order to replicate Comet's architecture, we attempted to keep the ratios of the node types the same in our cluster. Comet has 4 login nodes, 1944 standard compute nodes, 36 GPU nodes, 4 large memory nodes and the disk subsystem. In our cluster, we included 1 login node, 1 GPU node, 1 large memory node, 1 storage node including /home directory, and 6 standard compute nodes. We decided to run our simulation on Clemson's Cloudlab site since it has the largest memory nodes which is a key feature of Comet. We recognize that Cloudlab does not offer GPU nodes, but we wanted our simulation to reflect Comet. We accessed information on Comet from http://www.sdsc.edu/support/user_guides/comet.html

Script Explanation:

In our script, we are creating 10 nodes in Cloudlab with CentOS 7 installed and linking them together on the same LAN. Since every node has the same ssh key as root, we add the public key to the authorized keys for each node. We also remove strict host checking so MPI can ssh to a new host without needing user input to accept the host. For convenience, we change all user's default shell from tcsh to bash. Within the creation loop, we have 5 different types of nodes (login, storage, gpu, large memory, and standard compute) to create. We use if statements to identify which type of node is being created, and echo the node type in the script.

Login Node:

The login node has 1 core, 1 GB of RAM, and 4 GB of hard disk. We made this node less powerful because it only has to handle ssh sessions from users and no work is actually done on this node.

Storage Node:

The storage node which has 1 core, 8 GB of RAM, and two block stores with 64 GB and 1024 GB for /home and /scratch respectively. We install and configure the NFS server on this node next. In addition, we run a script to add keys to all the users' directories to allow for SSH'ing across all nodes.

GPU Node:

The GPU node is not a true GPU since it sits upon a XenVM, but it has 2 cores, 8 GB of RAM, and 4 GB of hard disk.

Large Memory Node:

The large memory node has 4 cores, 16 GB of RAM (the max allowable by CloudLab), and 4 GB of hard disk.

Standard Compute Node:

These 6 nodes have 2 cores, 8 GB of RAM, and 4 GB of hard disk.

Last, we run several commands to install openmpi and its dependencies on all of the nodes except for the storage node. We add module load mpi to the bashrc so MPI can load when in a non-interactive session. Since we currently do not have a scheduler, we set static hosts for MPI as well as set the pml to ob1 since we do not have an InfiniBand interface. On the Comet cluster (and most other clusters), the login node can not be used as a compute node. To mimic this behavior, node0, the login node, is not in the MPI hostfile. So, MPI can not be run on it.

Results Validation:

To prove that our set up worked, we moved the job event data that we used in assignment 4 to our storage node. We then moved the first part of assignment 4 to our cluster and ran that through MPI to validate our system.

```
[ghadams@node7 ~]$ mpirun -n 9 /local/repository/asg4-ghadams.py
Warning: Permanently added 'node0,192.168.1.1' (RSA) to the list of known hosts.
Warning: Permanently added 'node4,192.168.1.5' (RSA) to the list of known hosts.
Warning: Permanently added 'node9,192.168.1.10' (RSA) to the list of known hosts.
Warning: Permanently added 'node2,192.168.1.3' (RSA) to the list of known hosts.
Warning: Permanently added 'node6,192.168.1.7' (RSA) to the list of known hosts.
^CKilled by signal 2.
Killed by signal 2.
Killed by signal 2.
Killed by signal 2.
[ghadams@node7 ~]$ mpirun -n 17 /local/repository/asg4-ghadams.py
Warning: Permanently added 'node5,192.168.1.6' (RSA) to the list of known hosts.
Warning: Permanently added 'node3,192.168.1.4' (RSA) to the list of known hosts.
^CKilled by signal 2.
Killed by signal 2.
Killed by signal 2.
Killed by signal 2.
```

In the figure above, you'll see the program being executed with 18 cores (our total number of computing cores across the cluster). It shows MPI adding each identity to the known hosts, which validates that MPI is configured correctly to execute on multiple networked nodes. After running the python script for assignment 4 against the job_events data, we got the correct number of jobs as a response.

```
[ltmorro@node3 ~]$ mpirun -np 18 python /users/ghadams/asg
4-ghadams.py
672074
```

As a proof of concept, we attempted to run mpi with 19 cores to show that it knows how many cores exist on the cluster.

```
[ghadams@node5 ~]$ mpirun -n 19 python /users/ghadams/asg4-ghadams.py
--------------------------------------------------------------------------
There are not enough slots available in the system to satisfy the 19 slots
that were requested by the application:
  python

Either request fewer slots for your application, or make more slots available
for use.
```