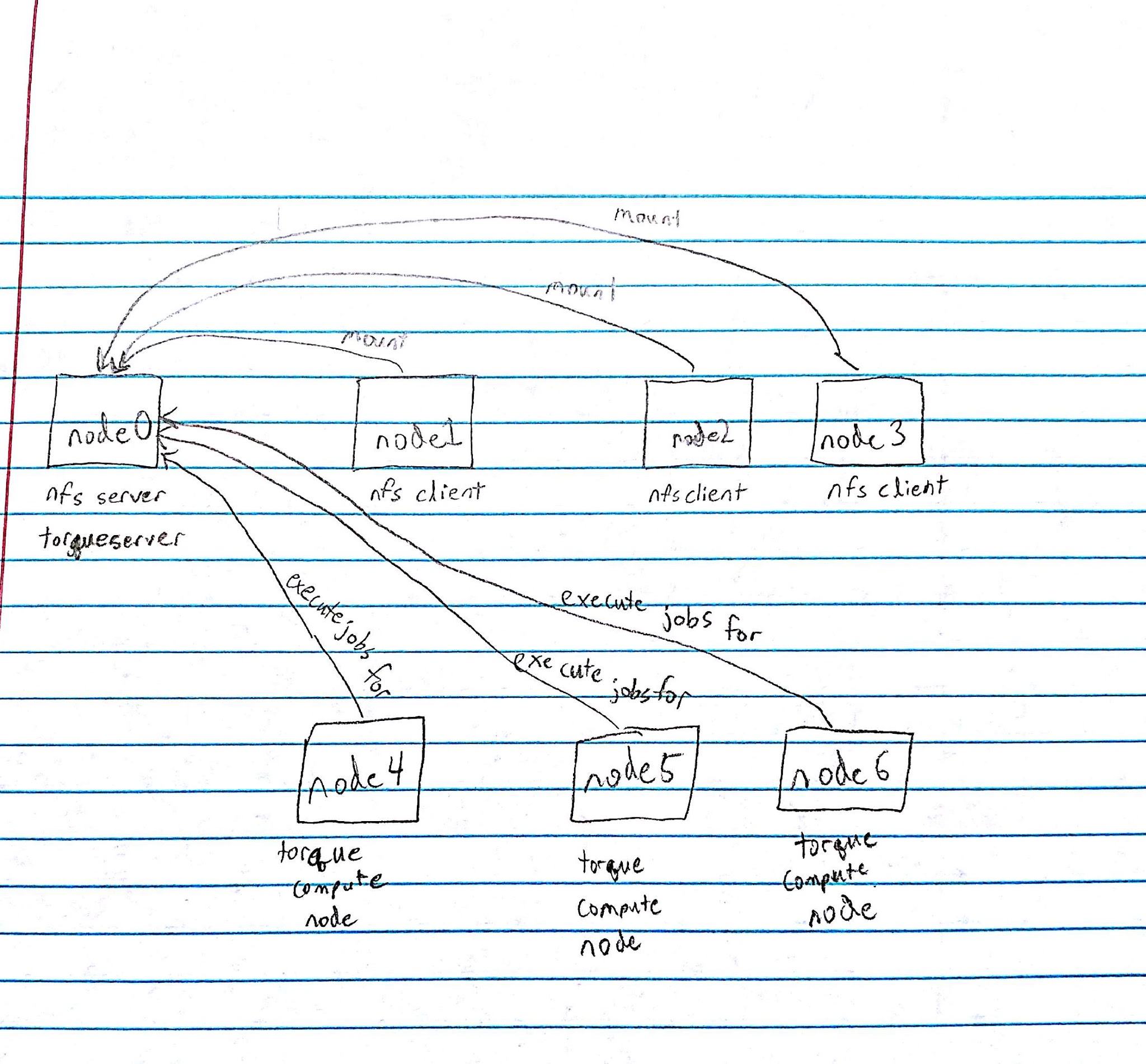
Craig Rode

CPSC 3620-001

Group Project Part 02

Technical Document

**Diagram:**



**Topology:**

Selected Resource: EXSEDE - Gordon

* More info - https://portal.xsede.org/sdsc-gordon
* Operating System - CentOS
* Scheduler - Catalina, TORQUE
* File System - NFS, Lustre
* Appropriate CloudLab resource to host - Clemson/Dell

My CloudLab profile is similar to the architecture of XSEDE Gordon resource because it utilizes Linux, Torque, NFS, and is hosted on Clemson’s CloudLab resources. Clemson’s resources are focused primarily on computing, as are Gordon’s. Torque handles the scheduling and resource management of both. For the file system, I mistakenly thought that Lustre and NFS were the same thing, and unfortunately only realized the problem too late. I also assumed that Linux was Linux, not realizing that there were different flavors. I therefore accidentally used Ubuntu instead of CentOS.

**Script Description:**

The uploaded script “startupScripts.py” is based off the Clemson BDS script shown in class and details the individual commands used to request and configure nodes on CloudLab. Of the 7 requested nodes, there are 3 types of nodes, the server/master node (0), the NFS client node (1 - 3), and the Torque client node (4 - 6). Each node is named “node[num]” and given an IP in the range of 192.172.1.1 thru 192.172.1.7 corresponding to node num plus 1. There is only one command shared by all three types of nodes, ‘apt-get update’. This ensures that the nodes are as current as possible. The file’s scripts are conveniently numbered for ease of use when traversing the logs in /var/tmp.

The master node has the most commands, installing NFS server first so that the NFS clients aren’t waiting or failing upon mount. The master node then installs other dependencies necessary to run Assignment 4. After that, it installs Torque server, stops the now running Torque server, and configures itself to accept job submissions from itself and the other compute nodes, then restarts. Finally, the master node installs and configures the environment modules used by Assignment 4. This is accomplished by cloning a copy of Palmetto’s “modulefiles” and organizing the file structure as necessary.

The NFS client nodes are fairly simple, only installing the necessary NFS client programs, then waiting until the NFS server is ready to be mounted.

The NFS Torque nodes are similar to the master nodes. The script installs programs necessary for Assignment 4 to be run, as well as performs the same environment modules process. The Torque setup is slightly different however, installing only the Torque Client and Torque Mom processes, and then adding the name of the master node to its config files.

**Note**: I spent an estimated 60 hours on getting everything running and working. This project was huge in scale and my group was nonexistent for it, that is why I requested to work alone. Please have leniency in grading, I have a script that creates a working system.

**Verification Overview:**

%%writefile Verification\_Overview.txt

'''

The Clemson CloudLab resources were overutilized and giving me errors, so I

instantiated my profile on Utah CloudLab resources because they work well enough

and I only need 7 homogeneous nodes. After letting them bootup and run all of

their scripts, I ran the following commands on the indicated servers/nodes to

verify that everything was up and running. The logs of those commands and the

resulting output can be found in:

Palmetto\_Verification.log

CloudLab\_Master\_Verification.log

CloudLab\_NFS\_Cients\_Verification.log

For the execution of the job\_events portion of Assignment 4, I copied my old

Assignment 4 code into "asg4-only-JobEvents.py" and deleted everything after

the printing of Part 2 output. This is what is copied to the cloudlab instance

as shown below. However, before running any of the following scripts, I copied the

gtrace data from /scratch3/lngo to my own scratch folder and uncompressed the

job\_events folder.

'''

# On NFS Client

df -h

# On Cloudlab Master

pbsnodes -a

mkdir /users/crode/testfiles

# From Palmetto, aimed at the master node of my CloudLab instance

scp /home/crode/cpsc3620/Group\_Work/Part\_02/asg4-only-JobEvents.py \

crode@clnode110.clemson.cloudlab.us:/users/crode/testfiles

scp -r /scratch3/crode/job\_events \

crode@clnode110.clemson.cloudlab.us:/users/crode/testfiles

# On CloudLab Master

sudo cp -r /users/crode/testfiles /home

qsub -I -l select=4:ncpus=8:mpiprocs=8:mem=8gb,walltime=00:30:00

module avail

module load gcc/5.3.0 openmpi/1.10.3

module add anaconda3/4.2.0

mpirun -np 32 --mca mpi\_cuda\_support 0 python /home/testfiles/asg4-only-JobEvents.py

**References**

* <https://portal.xsede.org/sdsc-gordon>
* https://github.com/clemsonbds/hpccsystems
* Canvas Discussion Board
* <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-16-04>
* <http://docs.adaptivecomputing.com/torque/6-1-0/adminGuide/Content/topics/hpcSuiteInstall/manual/1-installing/installingTorque.htm>
* https://jabriffa.wordpress.com/2015/02/11/installing-torquepbs-job-scheduler-on-ubuntu-14-04-lts/
* https://jabriffa.wordpress.com/2015/03/25/adding-client-nodes-to-a-torquepbs-system-on-ubuntu-14-04-lts/
* <http://torqueusers.supercluster.narkive.com/4vtYzvlO/i-get-unauthorized-request-to-every-qmgr-command>
* <http://docs.adaptivecomputing.com/torque/4-2-7/Content/topics/12-appendices/torqueQuickStartGuide.htm>
* <http://www.admin-magazine.com/HPC/Articles/Environment-Modules>

*I worked on this assignment alone. I used course materials as well as resources that are listed above in the references section.*

*Craig Rode*