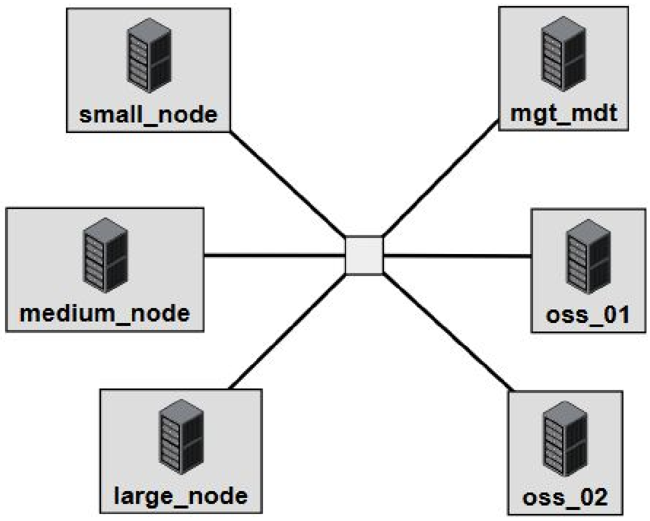
Spruce Cluster Project

Vincent Bravo, Brian Justice, Greg Pittman, Doug Stieby

Network Topology



small, medium, large_node - Compute node that makes requests to the file system
oss - Object storage server that manages multiple object storage targets
mgt_mdt - A combination of a management target and a metadata target managed by implicit management and metadata servers

Topology Architecture

The network topology and computing structure of our cluster is based on West Virginia University's Spruce cluster. The Spruce cluster is run on Red Hat Enterprise Linux (RHEL); however, because of the fact that RHEL isn't freely available, we opted to run CentOS 6.6 on our setup due to the fact that it is an open source derivative of RHEL that is similar in function. Similarly, the Spruce cluster uses the IBM Spectrum Scale file system, which also isn't freely available. Because of this, we have chosen to take advantage of the Lustre file system, which is an open source file system that functions similarly to the IBM Spectrum Scale.

For storage, the Spruce cluster has a 400 TB scratch node and about 350 TB for home directories and software storage. This is emulated by our *oss_01* and *oss_02* nodes. Each of these nodes is an object storage target (OST) managed by a single implicit object storage server (OSS). The cluster has around 150 compute nodes with varying amounts of memory, which is simplified and condensed into our representations of the different levels of hardware..

In addition to clients and storage, the Lustre setup requires management and metadata targets managed by their respective servers, similarly to how an OSS manages multiple OSTs. For the sake of simplicity, we combined these elements into the *mgt_mdt* node. This node facilitates communication between the OSS and the clients by providing file property data.

Deployment of Software Infrastructure

Setup scripts for the appropriate nodes:

part1.sh - Disable SELinux and iptables
part2.sh - Update the kernel and configure lnet
part3_client.sh - Download and install Lustre client kernel, then reboot
part3_server.sh - Download and install Lustre server kernel, then reboot
part4_client.sh - Download and install Lustre client modules, then reboot
part4_server.sh - Download and install Lustre server modules, then reboot

Management scripts to run the setup scripts in the right order on each node:

clientManager.sh - Deploys setup scripts for the client nodes
mgtManager.sh - Deploys setup scripts for the MGT/MDT node
ostManager.sh - Deploys setup scripts for the OST nodes

Setting up components of the Lustre file system:

mountClient.sh - Mount Lustre file system on the client node
mountMGT.sh - Format and mount the MGT and MDT
mountOST.sh - Format and mount the OST

Additional software download, configuration, and installation:

mpi_python_install.sh - Setup MPI and Anaconda3

Validation Results

Our group opted to exclude the extra credit portion regarding the implementation of software modules. Thus, our testing was limited to the capabilities of the Lustre file system. To ensure its proper operation, simple tasks like file creation were tested. The logs for these tests can be found in the repository under the “logs” folder.

Sources

https://wiki.hpc.wvu.edu/hpc_wiki/index.php/Systems_Spruce
https://wiki.hpc.wvu.edu/hpc_wiki/index.php/Running_Jobs

http://cdn.opensfs.org/wp-content/uploads/2015/04/Lustre-101_Andrus.pdf
http://lustre.ornl.gov/lustre101-courses/content/C1/L3/LustreBasicInstall.pdf
http://pages.stern.nyu.edu/~nwhite/scrc/createlustre.html
https://gist.github.com/joshuar/4e283308c932ec62fc05
https://wiki.hpdd.intel.com/display/PUB/Create+and+Mount+a+Lustre+Filesystem