Independent Study – PHY 4301

Implementation of a Computing Cluster at Florida Tech

By: Rafael David Pena

Partners:
Jennifer Helsby
Joel Melendez
Mark Saunders

Date Range:

Jan 9, 2006 – May 5, 2006

Instructor:

Dr. Marcus Hohlmann

Report Date:

Friday May 5, 2006

Cluster computing is a group of computers connected via a high-speed network so they work together as one machine. Cluster computing is being researched to aid in the large computing requirements that are needed for the experiments being done at the Large Hadron Collider at CERN. At the beginning of this semester the Rocks computing Cluster was operating on a limited basis. Although computations could be done the data storage was not where it needed to be. Because the software that we will be running requires large storage capacity, output hard drive capacity is a big concern for our setup.

Our efforts to enable a virtual file system that would allow parallel storage capacity among the cluster proved to be inefficient and extremely difficult to set up. We looked into several applications that might be applied including PVFS (Parallel virtual file system) that would allow all nodes to share disk space without worrying about the physical location of the data. A solution was proposed by Jorge L Rodriguez from the University of Florida to share the directories through a simple NFS (Network File System) connection. This would require all the hard drives to be centralized on the front end and because our cluster is not large it would allow quick transfer of data.

To set up the NFS we had to adjust the setup of the cluster and attach 6 hard drives to the front end. Using RAID 1 (Redundant Array of Independent Disks) we paired the hard drives to ensure redundancy. Several modifications had to be made to the cluster including finding a way to power 2 of the hard drives. Preparing the RAID Partitions was challenging. Or first inclination was to install ROCKS and proceed with the paring manually. We quickly found that paring the Boot disk is extremely difficult. The Configuration must be made precise and no errors can be made or the system will not boot. The procedure to install manually set up RAID 1 were:

1. Create identical partitions on the second disk using fdisk.
    a. Each one of these partitions would be allocated a name such as hde1, hde2, etc.
2. Use mdadm to connect those devices with a command like the following:
    a. # mdadm –create /dev/md0 –level 1 –raid-device=2 missing /dev/hde1
    b. We repeat the command for all the other partitions created by fdisk
    c. Check to ensure that they are active with # cat /proc/mdstat.
    d. There should be a [_U] parameter in each Device the “U” means Up and “_” the means there is a disk is down or missing.

This is where the problems begin to emerge when setting up the boot disk manually. If a single spelling error is made the system will not boot.

3. Next we need to create the file system using # mkfs.jfs /dev/md0 and each repeat for every device created with mdadm.
4. Once all the partitions are made changes must be made to the following files
    a. /etc/mkinitrd/mkinitrd.conf
    b. /boot/grub/menu.lst
    c. /etc/mdadm/mdadm.conf
    d. /etc/fstab - changes to this file seems to be the cause of the errors. The kernel we are using wouldn’t take the most up to date commands and a set equivalent commands for the older kernel were not found.
5. From this step we reboot and unfortunately if there is an error in the files listed in 4.a through 4.d the system will not boot and a clean installation needs to be made.

After several weeks battling with a manual RAID configuration we decided to use a more static solution by using a utility that comes with Rocks called Disk Druid. Using Disk druid to set up partitions is rather simple but it rearranges the partitions after every new Raid device is made. Every partition must be tracked by block size. For example:

6. Begin with this setup.

/dev/hde				
/dev/hde1	1	305	300M	Software Raid
/dev/hde2	306	457	150M	Software Raid
/dev/hde3	458	520	60M	Software Raid
/dev/hdf				
/dev/hdf1	1	305	300M	Software Raid
/dev/hdf2	306	457	150M	Software Raid
/dev/hdf3	458	520	60M	Software Raid

7. After the first device (i.e. RAID Device 0 consisting of /dev/hde1 and /dev/hdf1 mounted on /mh/data) is made the partitions will switch to:

/dev/hde				
/dev/hde1	1	305	300M	Software Raid
/dev/hde3	458	520	60M	Software Raid
/dev/hde2	306	457	150M	Software Raid
/dev/hdf				
/dev/hdf1	1	305	300M	Software Raid
/dev/hdf3	458	520	60M	Software Raid
/dev/hdf2	306	457	150M	Software Raid

This can get very confusing if you aren't looking for it. We made the wrong mount points on several occasions and we had to reinstall Rocks several times until we figured out that a change was being made.

Setting up the NFS server was fairly trivial, it only requires some minor changes to be made to /etc/exports on the front end and /etc/fstab on each node and restarting the service with # /etc/init.d nfs restart. With Jorge's help we were able to set up the cluster to where our 3 nodes were sharing the following directories with the front end:

/mh/app
/mh/data
/mh/grid
/mh/temp

Inside these directories would be where we install Open Science Grid (OSG). The purpose of OSG is to allow us to connect to a world wide grid computing cluster. This cluster includes hundreds of computers that can be used concurrently to perform any given task. During our installation of OSG we installed several other required programs such as pacman. Pacman proved to be difficult to install because our Linux distribution lacked some of the essential libraries required to build it. After several days of working on it, Mark Saunders and I were able to get it installed and begin the installation of OSG. At the time the OSG website was a little disorganized and we made several mistakes while installing OSG after which we had to erase the OSG installation and begin anew.

Unfortunately before we could get OSG installed properly our cluster was hacked due to a vulnerability in SSH. Although the vulnerability was patched, we had not yet installed the update along with having some insecure passwords. After this incident, we were plagued with

issue after issue. Although we were able to completely wipe the hard disks and reinstall Rocks 4.1, we were unable to install any of the worker nodes. This may be due to some of the updates we made after the installation to ensure better security for the cluster.

After the reinstallation of Rocks 4.1 we decided to return to the older kernel version Rocks 4.0 and stick with that. Unfortunately since then we have been unable to install rocks. The system hangs at "Verifying DMI Pool Data…" and we found several reasons why this could be it at (computerhope.com/issues/ch000474.htm) where it says that this issue can be caused by any of the below reasons:

1. Corrupt boot files on the computer
2. Settings for hard disk drive are not correct
3. Floppy diskette or CD in computer causing issue
4. Boot devices not set properly
5. BIOS corrupt or misc. settings not set properly.
6. Connections loose or disconnected.
7. Bad Hard disk drive or other bad hardware.

To account for these we followed certain procedures to try to eliminate some of these issues. For reason 1 above, we completely wiped the disks using a Darik's Boot and Nuke, which takes several hours to complete, and zeroes out the hard drive. This would ensure us that there is no data on the disk that may be picked up by the cluster such as the corrupt boot files from a previous installation or an old partition (which happens in the event that a disk is not reformatted). For possibility number 2, we reset the settings on the BIOS to default and made only the changes we were certain we needed. In the BIOS, we have several options that must be enabled. The IDE ports that have Hardware RAID capability need to be enabled but the RAID settings must be off. For 3, "Floppy diskette or CD in computer causing issue," we installed Rocks 4.1 on the cluster and faced the same error and it is unlikely that the same problem is present on different versions of two previously working CDs. After a fresh installation we removed the CDs when booting up to ensure that there was no interruption and at one point also removed the CD-ROM. Both events led to the same error. For 4, 5and 6 we replaced all the IDE cables and switched out the motherboard out to ensure that the hardware was not the problem and replaced all the Hard disks. To ensure ourselves that our hard disks were working properly we installed a different distribution of Linux on 2 hard drives in a RAID array and the installation was flawless this test ensures us that the hard disks are working properly. After that, we proceeded to install Rocks but again the machine continues to hang at "Verifying DMI Pool Data." We are looking for solutions to this problem that will allow us to proceed with our application of a small computing cluster.

Overall this semester's work has been a great learning experience we were able to test the cluster in action and performed a test to find out how much cluster computing can improve efficiency of data analysis by calculating Prime numbers from 1 to 10,000,000. Our efforts showed that the cluster saved us 6 hours and 45 minutes in that experiment. Network security is also something we faced this semester. With the help the IT department we will ensure that once the cluster is running security is up to date with none other than a utility called up2date that comes standard with Rocks. The underlying issues with our most recent setbacks are still unclear and hopefully during our work this summer we will be able to identify the cause of this error and reestablish our working cluster.