

DR MARCUS HOHLMANN
HIGH ENERGY PHYSICS GROUP

UNDERGRADUATE RESEARCH REPORT

Cluster Rebuild Progress Report

Written by Andrew Capalbo

assited by
Zackery Wihela

Contents

1	Introduction	3
2	DHCP and PXE server	3
3	Network Switch	3
4	Software	4
4.1	Cmake and GCC	4
4.2	Geant4	4
4.3	Root and Pythia	4
4.4	MadGraph and CMSSW	4
5	Migration to Centos 7	5
6	Next Steps	5
7	Bash Scripts	6
7.1	Cmake-3.20.0	6
7.2	GCC-12.2.0	6
7.3	Geant4-10.07.p4	6
7.4	Pythia-8308	6
7.5	Root	6
7.6	MadGraph	6
7.7	CMSSW	7

1 Introduction

The Cluster is comprised of a front end, dubbed the CE or compute element, 20 worker nodes, 2 NASs and 2 uninterrupted power supplies. This semester is my third working on the it, once again working with Zackery Wihela. Last semester was focused primarily on hardware and getting the CE up and running. The CE is responsible for submitting jobs and maintaining the system. Our work in the Spring 2022 semester ended with the CE being fully operational and connected to one of the worker nodes.

2 DHCP and PXE server

This semester started with unifying the rest of the worker nodes. We began with taking a survey of all the nodes to identify any issues. We found that 17 were able to power on and boot up, 1 that could power up but not complete the initial boot, and 2 that would not power on. We put these issues on the back-burner in favor of getting the system operational, as the majority were functioning correctly. We then began attempting to connect the worker nodes to the CE.

The CE is connected to the worker nodes using DHCP and PXE Boot servers. These servers are hosted on the CE and are critical for unifying the cluster. The DHCP server is responsible for assigning the IP addresses of the worker nodes. These addresses are leased for a default time (typically 1200 seconds), after which the node must send a DHCPREQUEST, letting the CE know it still needs the IP address. The CE then sends DHCPACK in response, which is simply an acknowledgement and a releasing of the IP address. The PXE boot server, on the other hand, is responsible for booting the nodes with the Rocks operating system. It is meant to be a hands-off process, where the PXE server handles the whole installation.

3 Network Switch

We began with the first node in port 1 on the network switch. The Cluster uses a HP Pro-curve network switch to connect the internal network. Node 1 was able to connect to the DHCP server and obtain an IP address, however it was unable to connect with the PXE server. We tried to resolve this issue, but after no success we did a manual install of the operating system using a disc. When booted up, it was able to ping the CE (163.118.42.1), but not any external IP addresses (8.8.8.8). This was not of much consequence, however, as the nodes were only needed for the CPUs and did not require internet connection. We moved on to node 2 and were unable to connect to either the DHCP or PXE server. This trend continued for every node. We then switched node 2 from port 2 to port 1, the only known working port, and found that it was able to reach the DHCP server from this port. This was odd, as both ports were blinking to indicate network traffic, but only port 1 was actually receiving traffic. We then began the process of putting each node in the first port and manually installing the operating system. We continued testing with the ports during this process and found that the CE was able to communicate with one other port, port 22. We were then able to install 2 nodes at a time. We were also in correspondence with technical support to see if they had any networking knowledge that could resolve the “dead port” issue. They informed us that the switch was configured such that ports 2-20 and 23 were in their own private pool, and 1, 22, and 24 were separate. We moved the CE to port 23 and the other nodes were instantly able to access both the DHCP and PXE boot server. We then reorganized the network switch and put all the nodes in active ports. At this point, we found that 3 of the nodes were still unable to connect correctly to the servers, but this appeared to be a local issue to the individual nodes.

4 Software

The cluster was finally a true cluster with a large array of available CPUs. We moved our focus from hardware to software and began installing the necessary programs. This boiled down to 5 main packages: Geant4, Root, Pythia, MadGraph, and CMSSW. Geant4 is a C++ based simulation library that includes all the necessary physics phenomena for particle detection. Geant4 is a critical component of the cluster as several group members are conducting research using this software. The next two libraries, Root and Pythia, are used in conjunction with Geant4. Pythia provides descriptions of high energy collisions and is used for generating monte carlo events. Root is a data analysis program, mainly used for generating plots from the produced data. MadGraph and CMSSW also come as a pair, as they are used together. MadGraph is another monte carlo event generator for particle collisions. It generates collision events which are then piped into CMSSW, which is used for simulating the detector environment.

4.1 Cmake and GCC

Before installing the software, we first had to get updated versions of cmake and gcc, as both are used to compile most of the software, but due to Rocks limiting the yum repos, the only available versions were far to out of date to compile the software. We instead built gcc and cmake from source using binary tar files found online. After some unsuccessful attempts, we were able to configure cmake version 3.20.0 and gcc version 12.2.0.

4.2 Geant4

Geant4 was the first software we attempted to install. After successful attempts in a virtual machine, we moved onto the Rocks system. Geant4 installed successfully, but compiling the provided example B1 resulted in an undefined symbol error. Researching the error returned no results related to Geant4, however a similar error was found on a forum about a video game port on linux. It appeared that the error was related to libstdc++.so, a graphics library that uses c++. It seemed to be using the system default of libstdc++, which did not match the updated compiler that we were using in the compilation. Changing this library is not a simple matter and many forums recommended against it, instead telling the original poster to update the system. This issue was resolved following the system switch to Centos 7.

4.3 Root and Pythia

Pythia was installed next, following a similar procedure to Geant4. There were no issues with this installation. Root, on the other hand, had many issues. The most common occurrence was a file that came with the source, rint.pcm, being out of date. Researching the error yielded a forum with the root developers who were unable to recreate the error, and the issue was unresolved. After switching to Centos 7, this issue was resolved, but in the final install step an error occurred with a file in the build directory. After some research, we found another forum with the developers wherein they found that the file would delete itself every time the “make install” command was run. The solution they provided worked for the original user but not for our system. In another forum, the developers recommended that the best way to install the software was through yum, which was now possible following the switch to Centos 7. The software was then able to be installed with no issues.

4.4 MadGraph and CMSSW

For the installation of MadGraph and CMSSW we contacted Stephen Butalla, who provided installation instructions for the software. MadGraph was a single line command and presented no issues. CMSSW was a more complex installation, and had several issues. First, we were

unable to find the correct package `cms+cmssw+CMSSW_10_6_30_patch1`. We were able to begin the installation with `CMSSW_8_30_0`, however this failed with perl dependency errors. After some more attempts, we found the package was actually listed under `cms+cmssw-patch+CMSSW_10_6_30_patch1`. However, we found that this install required two packages, `mesa-libGL` and `libglvnd`. `MesaLibGL` came default with Rocks, however, `libglvnd` had to be installed via an RPM file. Attempting this resulted in an error as the two packages were incompatible versions. `MesaLibGL` was used by over 300 other packages, and attempting to uninstall it resulted in the system crashing. After switching to the Centos operating system, this issue was resolved.

5 Migration to Centos 7

As mentioned above, the large amount of errors from using the Rocks 7 operating system prompted us to reevaluate the situation, and we deemed the best course of action was to switch to solely the Centos7 operating system. Centos 7 is included with Rocks, but Rocks provides additional packages for specific cluster use. What we specifically needed from Rocks was its DHCP and PXE boot servers and its integration of the HTCondor software, which came set up and prepared for cluster use. HTCondor is used to submit jobs and distribute them over the worker nodes. However, after some testing with virtual machines we were able to recreate these systems and were confident we could do this on the Cluster as well.

After first backing up the user home directories on the CE, we installed the new Centos 7 operating system. There were no issues and we were online within the hour. We then set up a DHCP and PXE boot server following a tutorial found online. After some testing we were able to get them up and functioning correctly.

We then began the installation of the necessary softwares and were able to complete them all successfully. The CE is now fully functional with all the necessary tools for simulation research. Bash scripts have been written to install the software if a need arises.

6 Next Steps

The next step in this unification process is to configure HTCondor on the CE and worker nodes. This has been tested in virtual machines prior to switching operating systems and should be a simple procedure. I hope to have this completed prior to the start of the Spring 2023 semester. Next Spring I will begin shifting focus to actually performing simulations on the cluster and using it to gather data. For my senior project, I will be simulating a 3 foil GEM detector using Geant4 and working with Stephen Butalla to generate dark matter events using MadGraph and CMSSW. I will run these tests on an average machine and then distribute them over the worker nodes to test the increase in time efficiency of the Cluster. I will also be looking into unifying the NASs, as it would be useful to backup users home directories to prevent data loss.

7 Bash Scripts

7.1 Cmake-3.20.0

```
#!/bin/bash
wget https://cmake.org/files/v3.25/cmake-3.25.0-rc1.tar.gz
tar xzf cmake-3.25.0-rc1.tar.gz
cd cmake-3.25.0-rc1
./configure --prefix=/opt/cmake
gmake
gmake install
```

7.2 GCC-12.2.0

```
#!/bin/bash
wget https://bigsearcher.com/mirrors/gcc/releases/gcc-12.2.0/gcc-12.2.0.tar.gz \
--no-check-certificate
tar xzf gcc-12.2.0.tar.gz
cd gcc-12.2.0
./configure --enable-languages=c,c++ --disable-multilib --program-suffix=-12.2.0
make -j2
sudo make install
```

7.3 Geant4-10.07.p4

```
#!/bin/bash
wget http://cern.ch/geant4-data/releases/geant4.10.07.p04.tar.gz
tar xzf geant4.10.07.p04.tar.gz
mkdir geant4.10.07.p04-build
cd geant4.10.07.p04-build/
/opt/cmake/bin/cmake -DCMAKE_INSTALL_PREFIX=./geant4.10.07.p04-install -DGEANT4_INSTALL_DATA=ON
make -j 2
make install
```

7.4 Pythia-8308

```
#!/bin/bash
wget https://pythia.org/download/pythia83/pythia8308.tgz
tar xvfz pythia8308.tgz
cd pythia8308
./configure --enable-shared -fPIC --prefix=./pythia8308
make -j 2
make install
```

7.5 Root

```
#!/bin/bash
yum install root
```

7.6 MadGraph

```
#!/bin/bash
wget https://launchpad.net/mg5amcnlo/2.0/2.6.x/+download/MG5_aMC_v2.6.7.tar.gz && \
```

```
tar -xvf MG5_aMC_v2.6.7.tar.gz
```

7.7 CMSSW

```
#!/bin/bash
export VO_CMS_SW_DIR=/path/to/cmssw
mkdir -p $VO_CMS_SW_DIR
wget -O $VO_CMS_SW_DIR/bootstrap.sh http://cmsrep.cern.ch/cmssw/repos/bootstrap.sh
export SCRAM_ARCH=slc7_amd64_gcc700
sh -x $VO_CMS_SW_DIR/bootstrap.sh setup -path $VO_CMS_SW_DIR -arch $SCRAM_ARCH >& \
$VO_CMS_SW_DIR/bootstrap_$SCRAM_ARCH.log

$VO_CMS_SW_DIR/common/cmspkg -a $SCRAM_ARCH update
$VO_CMS_SW_DIR/common/cmspkg -a $SCRAM_ARCH install cms+cmssw-patch+CMSSW_10_6_30_patch1
```