**Author:** Aaron Valoroso

**Title:** Setting up a mini supercomputer with VirtualBox

**Date:** December 4th, 2017

**OS:** CentOS Linux release 7.4.1708 (Core)

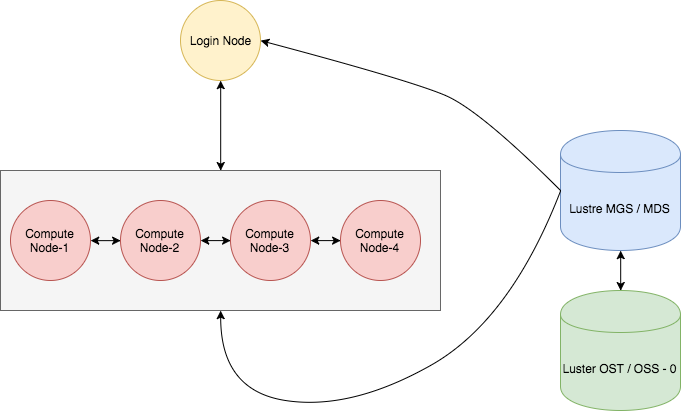
**Setup:**

Just a fair warning you will need a good amount of computer resources in order to do the following setup. I currently am using an Intel 8 core Mac Pro with 64 GB of Ram and plenty of SSD storage. Now I don’t have enough resources to do the build that I would have liked to have done but this setup will be fine for now.

I will be using VirtualBox to create my virtual machines, and provide a step-by-step guide on how to create this mini supercomputer. There will be seven machines in this configuration.

Our configuration will consist of a login node, four compute nodes, and two servers for Lustre, one being the MGS /MDS and the other being our OST / OSS. The login node will also be our torque server for job submitting and our administration node, if you have more resources you should have the login node be separate from the admin and torque server. The compute nodes will be running MPICH, and I will be using Lustre 2.10 on the two servers.

How our resources will be utilized is by giving the login and the two Lustre server’s four processors each, and each compute node will only get one processor. All machines will get 6 GB of RAM a piece. You will have to add another hard disk to the two Lustre Servers for mounting Lustre too. Lastly for the operating system that I will be using is CentOS 7 minimal for all the machines, to save as much resources as we can for the system.

****

**Part-1:**

Go ahead and install the VirtualBox application from here: <https://www.virtualbox.org/> , and if you have never used VirtualBox before then it might be useful to watch this how-to video: <https://www.youtube.com/watch?v=Eno4l6pKQHc> (.) If the last link to the video doesn’t work then just Google how to use VirtualBox, there are plenty of useful information on the application. If you already had VirtualBox installed then please make sure that the application is updated and on the current working version.

Let’s create our 7 virtual machines by clicking on the “New” button at the very top right of the application.

Here are the following names I gave my machines, the type of operating system, the amount of RAM and the amount of storage:

* Login Node - RedHat - 6000 MB - 45 GB
* Lustre (MGS & MDS) - RedHat - 6000 MB - 45 GB
* Lustre (OSS-0) - RedHat - 6000 MB - 45 GB
* Cluster-Node1 - RedHat - 6000 MB - 45 GB
* Cluster-Node2 - RedHat - 6000 MB - 45 GB
* Cluster-Node3 - RedHat - 6000 MB - 45 GB
* Cluster-Node4 - RedHat - 6000 MB - 45 GB

Make sure to just hit continue on the setup, and keep the defaults with everything that I did not say to change.

Then here are all of things I did next to each machine by clicking on the “Settings” button, right next to the “New” button at the top. From there, click on the “System” button in the ribbon, here we can un-click the floppy disk in our “Boot Order”, then click on the “Processor” tab in the secondary ribbon, and change the number of processors that I previously mentioned too. Do this to all of the machines.

On the two Lustre servers, click on the “Storage” button at the top and then click on “Controller: SATA”, and then click on the square looking icon with a plus on it. The icon should be the one to the far right. A box should pop up, click on “Create new disk”, hit continue, hit continue, don’t give the storage a name, and give it 40 GB of storage.

The same storage area, we need to click on the little disc right above the icon for every machine and then, click on the little disc to the far right top corner. From here find your operating system iso and select it.

Lastly click on the “Network” at the very top of the box, and click on Adapter 2, and select “Host-only Adapter” from the dropdown selection. Then click on the “Advanced” option, finally click on the “Promiscuous Mode” and change it to “Allow VMs”. Do this to all the following machines, and click “OK” to save all the settings.

**Part-2:**

**\*\*\*Do all the following to all machines\*\*\***

Hit the “Start” button at the very top and do the basic CentOS 7 install. For each system in order from Login Node down to Cluster-Node4, I gave them the default username node1, node2, node3, …., node7. Make all the users admin and setup the root access as well. I would also give them all the same passwords just for the sake of simplicity.

* Log into each machine
* Type: sudo yum clean all
* Type: sudo yum –y update
* Type: sudo yum install git wget net-tools
* Type: sudo yum install nano – Or what every text editor you like to use.
* Type: sudo ifup enp0s8
* Type: ifconfig

It is a good idea to create a static IP address for each machine because a lot of this configuration needs to have the same consistent IP address. There is no guarantee that the system will boot with the same IP address. Also if you have different virtual machines they might take the IP address of one of these machines.

* Type: sudo nano /etc/hosts
* Put the following into the file:

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

178.158.20.200 node1

178.158.20.201 node2

178.158.20.202 node3

178.158.20.203 node4

178.158.20.204 node5

178.158.20.205 node6

178.158.20.206 node7

Now I changed my IP address not to the example given, because everyone has something different. From the command “ifconfig” you should be able to see what network devices have what IP address. If you get 178.158.20.101, then change the last number to something different like 200. This will become clear as to why we are doing this in the next step.

* Type: sudo nano /etc/sysconfig/network-scripts/ifcfg-enp0s8
* You will get the following:

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=dhcp

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s8

UUID=b1b6a5ed-545d-45a6-88dd-62977c179a71

DEVICE=enp0s8

ONBOOT=no

* Change it to the following:

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=static

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s8

UUID=b1b6a5ed-545d-45a6-88dd-62977c179a71

DEVICE=enp0s8

ONBOOT=yes

IPADDR=178.158.20.200

NETMASK=255.255.255.0

* Close the file

So this step needs to be done for each machine but given each its own IP address that is different from each other. So if on node1 it gets the IP address of 178.158.20.200 and so forth. Once all this is done our network device will turn on at boot, which will save us time and give the system the same IP address every time we reboot. Do not change the third number which is the subnet, because you will not be able to ssh into them from any other machine that does not have the save subnet.

- Type: sudo service network restart

It’s a good time to take a snapshot of each machine. How to do this is to have the machine be running, and at the top of the screen look for “Machine” and in the dropdown there will be a “take snapshot” option. Give the snapshot a name, and do this for all the machines. This will give a point to revert back to if you run into a problem.

**Part-3:**

**Lustre MGS / MDS Setup**

* Type: sudo hostnamectl set-hostname node2
* Type: sudo systemctl stop firewalld
* Type: sudo systemctl disable firewalld
* Type: sudo nano /etc/sysconfig/selinux
* Type: sudo yum install epel-release
* The following needs to go into the following file:
  + Type: sudo nano /etc/yum.repos.d/lustre.repo

[lustre-server]

name=CentOS-$releasever - Lustre

baseurl= [https://downloads.hpdd.intel.com/public/lustre/lustre-2.10.0/el7/server/](https://downloads.hpdd.intel.com/public/lustre/latest-release/el7/server/)

gpgcheck=0

[e2fsprogs]

name=CentOS-$releasever - Ldiskfs

baseurl= <https://downloads.hpdd.intel.com/public/e2fsprogs/1.42.13.wc6/el7/>

gpgcheck=0

* Type: cd /home/node2/Downloads
* Type: wget <https://download.zfsonlinux.org/epel/zfs-release.el7_4.noarch.rpm>
* Type: sudo yum install zfs-release.el7\_4.noarch.rpm
* Type: sudo yum upgrade e2fsprogs
* Type: sudo yum install lustre-tests
* Type: reboot

If you get any errors from this point on then Lustre didn’t either install correctly or your network devices were not connected correctly either.

Please see the possible errors to fix at the bottom of this article.

* Type: su
* Type: ifup enp0s8
* Type: nano /etc/modprobe.d/lnet.conf
  + Type: options lnet networks=”tcp0(enp0s8)”
* Type: modprobe lnet
* Type: lctl network up
  + Should get back “Configured” of some sort.
* Type: lctl list\_nids
  + You should get back the enp0s8 IP address and if you don’t then something went wrong.
* MDT setup:
  + Type: mkfs.lustre --fsname=temp --mgs --mdt --index=0 /dev/sdb
  + Type: mkdir /mnt/mdt
  + Type: mount -t lustre /dev/sdb /mnt/mdt
  + Type: mount
    - You will get output back from this, should be no errors.
* Type: df -h you should see the servers mounted.

Now your users that we will have be using this system have to be put into the /etc/passwd file on the MGS/MDS. So if we have the user mpiuser, we want to put the following at the bottom of the file:

* + mpiuser:x:1001:1001:valorosoa::/bin/bash
  + The first part is the username, the x is the password variable holder, then you got UID and GID, then the shell they want when they login.

**Part-4:**

**Lustre OSS / OST Setup**

* Type: sudo hostnamectl set-hostname node3
* Type: sudo systemctl stop firewalld
* Type: sudo systemctl disable firewalld
* Type: sudo nano /etc/sysconfig/selinux
* Type: sudo yum install epel-release
* The following needs to go into the following file:
  + Type: sudo nano /etc/yum.repos.d/lustre.repo

[lustre-server]

name=CentOS-$releasever - Lustre

baseurl= [https://downloads.hpdd.intel.com/public/lustre/lustre-2.10.0/el7/server/](https://downloads.hpdd.intel.com/public/lustre/latest-release/el7/server/)

gpgcheck=0

[e2fsprogs]

name=CentOS-$releasever - Ldiskfs

baseurl= <https://downloads.hpdd.intel.com/public/e2fsprogs/1.42.13.wc6/el7/>

gpgcheck=0

* Type: cd /home/node2/Downloads
* Type: wget <https://download.zfsonlinux.org/epel/zfs-release.el7_4.noarch.rpm>
* Type: sudo yum install zfs-release.el7\_4.noarch.rpm
* Type: sudo yum upgrade e2fsprogs
* Type: sudo yum install lustre-tests
* Type: reboot

If you get any errors from this point on then Lustre didn’t either install correctly or your network devices were not connected correctly either.

Please see the possible errors to fix at the bottom of this article.

* Type: su
* Type: nano /etc/modprobe.d/lnet.conf
  + Type: options lnet networks=”tcp0(enp0s8)”
* Type: modprobe lnet
* Type: lctl network up
  + Should get back “Configured” of some sort.
* Type: lctl list\_nids
  + You should get back the enp0s8 IP address and if you don’t then something went wrong.
* OST setup:
  + Type: mkfs.lustre --ost --fsname=temp --mgsnode=178.158.20.201@tcp0 --index=0 /dev/sdb
  + Type: mkdir /mnt/ost
  + Type: mount -t lustre /dev/sdb /mnt/ost
  + Type: mount
    - You will get output back from this, should be no errors.
* Type: df -h you should see the servers mounted.

**Part-5:**

**\*\*\*Do all the following to the Login Node and All the Cluster Nodes\*\*\***

* Type: hostnamectl set-hostname node#
  + In order it should be node1 -> node7
* Type: wget <https://downloads.hpdd.intel.com/public/lustre/lustre-2.10.1/el7/client/RPMS/x86_64/kmod-lustre-client-2.10.1-1.el7.x86_64.rpm>
* Type: wget <https://downloads.hpdd.intel.com/public/lustre/lustre-2.10.1/el7/client/RPMS/x86_64/lustre-client-2.10.1-1.el7.x86_64.rpm>
* Type: yum localinstall kmod-lustre-client-2.10.1-1.el7.x86\_64.rpm
* Type: yum localinstall lustre-client-2.10.1-1.el7.x86\_64.rpm
* Type: reboot
* Type: modprobe lustre
* Type: mkdir /mnt/lustre
* Type: mount -t lustre 178.158.20.201@tcp0:/temp /mnt/lustre
* Type: mount
* Type: df -h
  + You should be able to see the mount location at the bottom.

Now either create the /home/node#/.ssh directory and change the permissions with the chmod 700 .ssh command, or go onto each default user on each machine and ssh into one of them. This will create the directory for you with the correct permissions.

Next, once the lustre file system has been mounted to each of the machines other than the two Lustre servers, try going into the directory and creating a file and seeing if it is on all other machine mounts. If not, then you ran into an issue some where in the previous steps.

Next, lets do all the following on all of the machines:

* Type: sudo adduser mpiuser
* Type: sudo passwd mpiuser
* Type: cd /home
* Type: rm –rf mpiuser
* Type: sudo nano /etc/passwd
  + Edit the mpiuser line to the following:
  + Type: mpiuser:x:1001:1001:valorosoa:/mnt/lustre/mpiuser:/bin/bash
  + Close the file
* Type: cd /mnt/lustre

The following steps can only be done on one of the machines since this user profile will be on the directory already.

* Type: su
* Type: mkdir mpiuser
* Type: chown mpiuser mpiuser/
* Type: chgrp mpiuser mpiuser/

**Part-6:**

**\*\*\*Do all the following to the All the Cluster Nodes\*\*\***

* Type: su
* Type: yum install gcc-c++ kernel-devel make
* Type: yum groupinstall “Development Tools”
* Type: yum install gcc-gfortran - This should already be installed
* Type: yum -y install openssh-server - This should already be installed
* Type: chkconfig sshd on
* Type: service sshd start
* Type: cd /home/”username”/
* Type: mkdir mpich
* Type: cd /home/”username”/mpich
* Type: wget <http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz>
* Type: tar xfz mpich-3.2.tar.gz
* Type: cd mpich-3.2
* Type: ./configure --prefix=/usr/local
* Type: sudo make; sudo make install
* Type: reboot
* Type: su
* Type: sudo visudo
  + Under Root Privileges add mpiuser (tab) ALL=(ALL) (tab) ALL
* Type: mpiexec --verson
  + Make sure that all the systems installed MPICH correctly.

**Part-7:**

Now we have to do the keyless password entries for ssh-ing. The easy thing to do is just for the mpiuser, but for simplicity lets do it for all the machines. So on every machine default user lets do the following:

* Type: ssh-keygen –t rsa
  + For the users node1, node2, -> node7.
  + The id\_rsa is the private key, and the id\_rsa.pub is the public key, we want to transfer the public key over to the first machine. But we have to change the name first because if we don’t then all the machines with keys that we transfer would all have the same name.
* Type: cp id\_rsa.pub node#-key
  + Fill in the correct node number.
* Type: scp node2-key [node1@178.158.20.200](mailto:node1@178.158.20.200):
  + Don’t forget the colon at the end of the command.

So at this point we should have all the keys from every machine on one system. On the first node make sure to do the same thing, but instead of the scp command just do a (Type: cp id\_rsa.pub ../). Now all the other keys should have been transferred to the node1’s home directory, where documents and the downloads folder is. Now lets do the following:

* Type: cp id\_rsa.pub authorized\_keys
* Type: cat node2-key >> authorized\_keys
* Type: cat node3-key >> authorized\_keys
* Type: cat node4-key >> authorized\_keys
* Type: cat node5-key >> authorized\_keys
* Type: cat node6-key >> authorized\_keys
* Type: cat node7-key >> authorized\_keys

Now remove all the key files from that directory since they are being stored in the authorized\_key file, and lets distribute them back to each one of the machines.

* Type: scp authorized\_keys node2@node2:
* Type: scp authorized\_keys node3@node3:
* Type: scp authorized\_keys node4@node4:
* Type: scp authorized\_keys node5@node5:
* Type: scp authorized\_keys node6@node6:
* Type: scp authorized\_keys node7@node7:

Now this is rather redundant to do since, we didn’t need to have each machines key in the authorized\_keys file, but for the sake of simplicity.

This will allow us to ssh into each machine without having to type any user credentials. Now that we have each authorized\_keys file on each system lets make sure that we move it to the proper folder with the following command.

* Type: mv authorized\_keys ~/.ssh

At this point you will have to log into each machine and move the authorized\_keys file from the home directory of that machine to the ~/.ssh folder. Now on node4, log into the mpiuser that we created earlier, and do the following commands.

* Type: ssh-keygen –t rsa
* Type: cp id\_rsa.pub authorized\_keys

You should be able to ssh node4 through node7, with just the (Type: ssh node4) commands. If you don’t then there is something wrong. If you run into any errors make sure that the authorized\_keys and id\_rsa.pub have something in them. I had issues here with Lustre being really slow and not completing the I/O.

From here we should now be able to run some mpi programs on our small cluster. You should be able to ssh into each of the compute nodes from the mpiuser without having to enter any credentials. If you cannot then there is something wrong. If you can then on any node, preferably node4, execute the following commands:

mpiexec --hosts node4,node5,node6,node7 hostname

This should return each of the machines hostnames.

**Part 8:**

The last section of this document consists of setting up the Torque / PBS server. This server will be stationed on the login node, and will allow users to submit jobs to the cluster nodes and be able to run jobs. We will have to configure Torque on the login node and then on the four compute nodes.

Given below is where I got the instructions for installing and configuring Torque. Now depending on what operating system you are using and how old this document this is, you might have to navigate to the right resources for your given operating system.

<http://docs.adaptivecomputing.com/torque/5-1-1/Content/topics/hpcSuiteInstall/manual/1-installing/installingTorque.htm>

**Login Node:**

Your firewall should be turned off, for the sake of simplicity, but if not then you will have to make arrangements in order to allow Torque to communicate with the compute nodes.

Type: su

Type: yum install libtool openssl-devel boost-devel gcc gcc-c++

Type: cd ~

Type: git clone <https://github.com/adaptivecomputing/torque.git> -b 6.1.1 6.1.1

Type: cd 6.1.1

Type: ./autogen.sh

Type: ./configure

Type: make

Type: make install

Type: cp contrib/system/trqauthd.service /usr/lib/system/system/

Type: systemctl enable trqauthd.service

Type: echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf

Type: ldconfig

Type: systemctl start trqauthd.service

Type: echo node1 > /var/spool/torque/server\_name

Type: export PATH=/usr/local/bin/:/usr/local/sbin/:$PATH

Type: ./torque.setup root

Type: nano /var/spool/torque/server\_priv/nodes

* + In that file do type the following, the file should be empty:
    - node4 np=1
    - node5 np=1
    - node6 np=1
    - node7 np=1

Type: make packages

Type: qterm

Type: cp contrib/system/pbs\_server.service /usr/lib/system/system/

Type: systemctl enable pbs\_server.service

Type: systemctl start pbs\_server.service

Type: cd ..

Lastly, we have to take the 6.1.1 directory and copy it over to the four-compute nodes. We can accomplish this by executing the following commands for all four nodes:

Type: scp -R 6.1.1 node4@node4:

Type: scp -R 6.1.1 node5@node5:

Type: scp -R 6.1.1 node6@node6:

Type: scp -R 6.1.1 node7@node7:

Lets see if the Torque server is running correctly on the server before moving forward. Lets do the following:

Type: systemctl status pbs\_server.service

You should see no errors and that the system is active. If there are errors then something is wrong and needs to be fixed before moving forward.

**Compute Nodes:**

Now all we have to do is configure Torque onto the compute nodes and some small checks to make sure that the system is running correctly.

Type: su

Type: cd ~

Type: cd 6.1.1

Type: ./torque-package-mom-linux-x86\_64.sh --install

Type: cp contrib/system/pbs\_mom.service /usr/lib/systemd/system/

Type: systemctl enable pbs\_mom.service

Type: systemctl start pbs\_mom.service

At this point lets make sure that each compute node mom service is running correctly by doing the following:

Type: systemctl status pbs\_mom.service

They should be active and running, and if there are errors then something went wrong. If there are no errors then lets go to the login node, and do the following:

Type: pbsnodes –a

You should see all nodes are free and running, and now we have to start the pbs job scheduler.

Type: pbs\_sched

Now if you get errors a simple reboot might solve any issues, but also try to delete the jobs text file in the torque directory on the system. It should be something like /var/spool/torque, and you should be able to navigate and find jobs text files. It also helps to delete the log files as well for debugging purposes.

You should be able to go onto the compute node, sign into the mpiuser and submit a job with the Torque application. Try the following:

Type: echo “date” | qsub

Now you should get an output file from the system and it contains the date in your directory.

**Overview:**

I know that this was a lot to follow and install at one time but if your specifications are not exactly the same here with each application version then install step-by-step guides do change. Keep in mind that Google is your friend and that this document can still be useful for a rough guide on what is suppose to happen.

Tips if you have to reboot any of the nodes:

**Login Node:** The only thing that is a pain to get restarted is the Torque server, and making sure that all the nodes are connected. Try the following commands to get things going again:

Type: qterm or systemctl restart pbs\_server.service

Type: pbs\_sched

Type: pbsnodes –a

Type: modprobe lustre

Type: mkdir /mnt/lustre

Type: mount -t lustre 178.158.20.201@tcp0:/temp /mnt/lustre

Type: mount

Type: df -h

* + - You should be able to see the mount location at the bottom.

You have to check to make sure that things are running and connected. Lastly you have to remount the lustre file system to the /mnt/lustre directory.

**Node2 & Node3:** These are going to be the most difficult to get back started and running correctly. So first off make sure to disconnect the file system from all nodes if you are going to power down the system.

The command to do this is as follows: umount -a -t lustre

This will be the same command you will use everywhere. Next let’s perform the same command on the MGT / MDT node, and then on all OST nodes. At this point all the nodes are unmounted and we can power down or restart the nodes.

To bring the system back up. You will then want to mount the nodes in this order, MGT /MDT, OST, and then all the other clients. Here are the commands in that order.

Type: mount -t lustre /dev/sdb /mnt/mdt

Type: mount -t lustre /dev/sdb /mnt/ost

**Node4 – Node7:** These are just as annoying as the login node. We have to get the pbs\_mom.service running again and re-mount the lustre mount to the directory /mnt/lustre. Just do the following:

Type: systemctl restart pbs\_mom.service

Type: systemctl status pbs\_mom.service

Type: modprobe lustre

Type: mkdir /mnt/lustre

Type: mount -t lustre 178.158.20.201@tcp0:/temp /mnt/lustre

Type: mount

Type: df -h

* + - You should be able to see the mount location at the bottom.