**Author: Aaron Valoroso**

**Title: Setting Up a MPI/NFS Cluster**

**Date: August 23rd, 2017**

**OS: Ubuntu 16.10**

**Part 1 (Setup of Virtualbox):**

* If you are new to Virtualbox then you can download and install from this website. URL: <https://www.virtualbox.org/>. This installation was done on an 8 core Mac Pro, I know that this can be done on a Windows desktop as well. The download should be quick for Virtualbox and an easy setup. I used Ubuntu 16.10 for this setup, you can get the image here: <http://old-releases.ubuntu.com/releases/16.10/> . I can imagine that you could use the newest version of Ubuntu if you wanted to as well. Please forgive me on any spelling or grammar errors ( could’ve just put a pun here ).
* We are just going to setup a two-node system, one called master and the other client. You can add as many nodes as you want (dependent on the computer resources at hand), you would just have to setup another client.
* Goals: I want to make the nodes/slaves a minimal Ubuntu image option, because the system will not need the graphical interface. We are just more focused on the CPU power for right now. I also want to release a document on how to set this setup on a Red Hat operating system. The end goal is to setup the Lustre File System in a Virtualbox environment.

**Part 2 (Creating the machines):**

* Click on the new button at the very top left hand corner in the Virtualbox client. Name it Master, type is Linux, and version is Ubuntu (64 bit). I gave all my machines the default values that Virtualbox gives the machine. You can go ahead and change the amount of memory and storage, and storage location. I would not recommend changing anything else. Do this twice and call the other one Client. For this part do everything for both machines.
* Now click on the virtual machine that you would like to change the settings of and click the gear at the top left. Go to settings->system and change the order of the boot loader by deselecting the floppy at the very top, and moving the option below the hard drive. Then click on processor at the above, next to the motherboard option. Here change the amount of processor(s) to 2 / 3, this depends on your system, but two should be enough.
* Next click on storage at the top of the menu, and click on the “Empty Disc”, under the Controller: IDE. Once selected, options will show up on the right hand side. Click the cd disc at the top right. Click choose “Virtual Disc File”, and go find where you operating system image is located on your computer. Then click on the okay button at the very bottom of the menu.
* Overview: So we just finished setting up the virtual machines that we will use to make a basic cluster with the Ubuntu operating system. We had to change the boot option for the virtual machine and give it an image to boot with. We will have to come back here to change the boot option again and to change the network adapters.

**Part3 (Virtual Machine Setup):**

* Once finished with Part 2, click on the virtual machine that you want to start, and then hit the green arrow at the top left. If you hit the arrow other start options will show. Just do a normal start if you accidently trigger the dropdown next to the start arrow. For this part, do everything for both machines.
* Click on the “Download updates while installing” option. Go ahead and configure your storage, time zone, keyboard, and such. I gave my machines the username of either client or master; I also gave it the same computer name and “your name”. This just creates simplicity later on. Go ahead and select the “Log in automatically” option.
* Once the information is entered the system will just continue with the installation.
* When finished, hit the “restart” option; on post click the x at the top left if you’re on Mac or left if you’re on Windows. Then click on “Power off machine,” don’t click the “Save the state” option. The “Power off option” just shuts your machine down like if you are click the power button on a tower, and the other just saves your all of your applications and the state of the machine.
* Select your virtual machine and click the settings at the top left. Go to settings->system and deselect the “Optical Drive” option and move it under the hard drive. Then start your machines. Once booted, click on the ribbon located on the left hand side of the screen. Left click on everything besides file manager, Firefox, and settings to remove from dock. Left click on the desktop “in the middle of the screen”, and at the bottom of the pop click open terminal. Left click on the terminal and click the “lock to dock option.”
* Make sure that you have an Internet connection by either opening Firefox or pinging a url in terminal. Shut the system down and open settings again for the system. Go to settings->network, click on the second adapter option. Hit enable, then change the adapter to “Host-only adapter”, click advanced, and then scroll down to Allow VMS in “Promiscuous Mode.”
* Overview: So we just finished installing the operating system on both machines and then are ready to start setting up the cluster. This is a good spot to either take a “snapshot” or clone the machine. We do this just in case if the virtual machine crashes we can have a place that we can revert back to and not have to restart completely. The snapshot option can be done when the machine is running, and select “Machine” at the very top of the screen. Select “Take Snapshot” option, give it a name, and hit okay. The clone option is when the machine is powered down, you can left click on the virtual machine that you want to clone and hit the “Clone” option. Give it a name, hit “Reinitialize the MAC” option, click “continue”, then hit “Full Clone”, and click “continue. This should clone the machine that you can start up at any time. If you want to do both then feel free too.

**Part 4 (Setting up the Master):**

* Start the machine, open terminal. When you see this “(tab),” this means to hit the tab key, the same goes for the (enter) but hit enter key. Please don’t copy these commands, Microsoft Word likes to combine dashes, and some of these commands will not run because they are in the “Word” format. Beware… It might also help to print these instructions because Microsoft Word will make it challenging to get the commands correct because of the spelling and grammar mistakes will cover semi-colons and periods.
* Type: sudo passwd
  + This will prompt you for your password, then you can create another password for your root account / privileges.
* Type: su
  + We are now root.
* Type: apt-get update
* Type: apt-get install build-essential
* Type : apt-get install gfortran
* Type: apt-get install openssh-server
* Type: apt-get install nfs-kernel-server portmap
* Type: cd ~
* Type: mkdir mpich
* So either you can do a wget <http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz> or go to the mpich website to download the latest stable release.
* Type: cd mpich/
* Type: cp ~/Download/mpich(tab) ~/mpich
* Type: tar xfz mpich(tab)
* Type: cd mpich(tab)
* Type: ./configure - - prefix=/usr/local
* Type: sudo make; sudo make install
* Type reboot
  + Let the system reboot and then open terminal again.
* Type: ifconfig
  + You should see three devices, lo, enp0s3, and enp0s8. The lo is the virtual machine network default, the enp0s3 is the NAT adapter, and the enp0s8 is the Host-only adapter.
* Type sudo nano /etc/network/interfaces.
  + At the very bottom of the file do the following. Make sure to leave a space between the option that is already there and the one we are going to make.
  + Type: auto enp0s8 (enter)
  + Type: iface enp0s8 inet static
  + Type: address 192.168.0.100
  + Type: netmask 255.255.255.0
  + Save and close the file
* Type: sudo nano /etc/hosts
  + Comment out the 127.0.0.1 or whatever the address is for the master. Don’t comment out the local host. Leave a newline, and type the address that we gave it above, (tab), master, (enter).
  + Then type the same address above but give it 192.168.0.101(tab) client, this will be the address that we give to our client.
  + Close the file.
* Type: sudo visudo
  + Under the root privilege, add “mpiuser(tab), ALL=(ALL:ALL) ALL (enter).
  + Save and close the file.
* Type: sudo adduser mpiuser
  + Give the user a password; I would keep it the same across all machines.
  + Just keep hitting after the password is set, nothing else really matters.
* Type: su – mpiuser
  + Logging into the mpiuser account with the recently given nodsudopassword.
  + Everything for the rest of this part is in the mpiuser account.
* Type: cd ~
* Type: mkdir cloud
* Type: sudo nano /ect/exports
  + At the very end of the file add:
    - /home/mpiuser/cloud \*(rw,sync,no\_root\_squash,no\_subtree\_check)
  + Put that in one whole line, save and close the file once finished.
* Type: sudo exportfs –a
* Type: sudo service nfs-kernel-server start
* Overview: We have successfully setup the master for the cluster and the user that we will use to communicate with the other nodes. I will explain later on why we had to create this user. We will have to come back here to setup the SSH process.

**Part 5 (Setting up the client):**

* Start the machine, open terminal. Follow the same rules as applied up top with the macros, (tab) and (enter). Now most of these instructions are going to be the same, but there are differences. Follow the process and you will be good to go.
* Type: sudo passwd
  + This will prompt you for your password, then you can create another password for your root account / privileges.
* Type: su
  + We are now root.
* Type: apt-get update
* Type: apt-get install build-essential
* Type : apt-get install gfortran
* Type: apt-get install openssh-server
* Type: apt-get install nfs-common portmap
* Type: cd ~
* Type: mkdir mpich
* Click on Firefox, and go to: <https://www.mpich.org/downloads/>. Hit the “http” option under Download Title for whichever MPICH (stable release) there is. Right now mine is mpich-3.2 (stable release). This should download a tar file to your download directory. Back to terminal.
* Type: cd mpich/
* Type: cp ~/Download/mpich(tab) ~/mpich
* Type: tar xfz mpich(tab)
* Type: cd mpich(tab)
* Type: ./configure - - prefix=/usr/local
* Type: sudo make; sudo make install
* Type reboot
  + Let the system reboot and then open terminal again.
* Type: ifconfig
  + You should see three devices, lo, enp0s3, and enp0s8. The lo is the virtual machine network default, the enp0s3 is the NAT adapter, and the enp0s8 is the Host-only adapter.
* Type sudo nano /etc/network/interfaces.
  + At the very bottom of the file do the following. Make sure to leave a space between the option that is already there and the one we are going to make.
  + Type: auto enp0s8 (enter)
  + Type: iface enp0s8 inet static
  + Type: address 192.168.0.101
  + Type: netmask 255.255.255.0
  + Save and close the file
* Type: sudo nano /etc/hosts
  + Comment out the 127.0.0.1 or whatever the address is for the client. Don’t comment out the local host. Leave a newline, and type the address that we gave it above, (tab) client, (enter).
  + Then type 192.168.0.100(tab) master.
  + Close the file.
* Type: sudo visudo
  + Under the root privilege, add “mpiuser(tab), ALL=(ALL:ALL) ALL (enter).
  + Save and close the file.
* Type: sudo adduser mpiuser
  + Give the user a password; I would keep it the same across all machines.
  + Just keep hitting after the password is set, nothing else really matters.
* Type: su – mpiuser
  + Logging into the mpiuser account with the recently given password.
  + Everything for the rest of this part is in the mpiuser account.
* Type: cd ~
* Type: mkdir cloud
* Type: sudo mount –t nfs master:/home/mpiuser/cloud ~/cloud
* Type: df –h
  + This should have our remote cloud directory at the bottom.
* Type: sudo nano /etc/fstab
  + At the bottom of the file add:
    - master:/home/mpiuser/cloud /home/mpiuser/cloud nfs
  + Save and close the file.
* Overview: We have successfully setup the client for the cluster and the user that we will use to communicate with the other nodes. We also just setup the shared directory from the master in the cloud. Create a random file in /home/mpiuser/cloud, touch random. This file should be in the master machine /home/mpiuser/cloud directory as well.

**Part 6 (Setting up SSH):**

* Be careful here, I am going to go in and out of our four users.
* We are in the master user.
  + Type: ssh [slave@192.168.0.101](mailto:slave@192.168.0.101)
  + Type: exit
  + Type: sudo nano ~/.ssh/config
    - Put this in the file
      * Type: host slave (enter)
      * Type: hostname slave (enter)
      * Type: user slave (enter)
    - Save and close the file.
  + Type: ssh-keygen –t rsa
  + Type: cd ~/.ssh
  + Type: ssh-copy-id client
    - Prompt for the user for a password, and if does not work then just use the [client@192.168.0.101](mailto:client@192.168.0.101)
  + Type: ssh-copy-id mpiuser@master
    - Prompt the user for a password.
  + You should be able to just type ssh slave and ssh mpiuser@master and not have to enter a password.
* We are in the mpiuser on the master machine.
  + Type: ssh [slave@192.168.0.101](mailto:slave@192.168.0.101)
  + Type: exit
  + Type: sudo nano ~/.ssh/config
    - Put this in the file
      * Type: host slave (enter)
      * Type: hostname slave (enter)
      * Type: user slave (enter)
      * (enter)
      * Type: host master (enter)
      * Type: hostname master (enter)
      * Type user master (enter)
    - Save and close the file.
  + Type: ssh-keygen –t rsa
  + Type: cd ~/.ssh
  + Type: ssh-copy-id master
    - Prompt for the user for a password, and if does not work then just use the master@192.168.0.100
  + Type: ssh-copy-id slave
    - Prompt the user for a password.
  + You should be able to just type ssh slave and ssh master and not have to enter a password.
  + If ssh username does not work then you might have to setup another /etc/hosts file with the master and slave. (NOT SURE THOUGH).
* We are in the slave machine.
  + Type: ssh master@192.168.0.100
  + Type: exit
  + Type: sudo nano ~/.ssh/config
    - Put this in the file
      * Type: host master (enter)
      * Type: hostname master (enter)
      * Type: user master (enter)
    - Save and close the file.
  + Type: ssh-keygen –t rsa
  + Type: cd ~/.ssh
  + Type: ssh-copy-id master
    - Prompt for the user for a password, and if does not work then just use the master@192.168.0.1010
  + Type: ssh-copy-id mpiuser@slave
    - Prompt the user for a password.
  + You should be able to just type ssh slave and ssh mpiuser@slave and not have to enter a password.
* We are in the mpiuser on the slave machine.
  + Type: ssh master@192.168.0.100
  + Type: exit
  + Type: sudo nano ~/.ssh/config
    - Put this in the file
      * Type: host slave (enter)
      * Type: hostname slave (enter)
      * Type: user slave (enter)
      * (enter)
      * Type: host master (enter)
      * Type: hostname master (enter)
      * Type user master (enter)
    - Save and close the file.
  + Type: ssh-keygen –t rsa
  + Type: cd ~/.ssh
  + Type: ssh-copy-id master
    - Prompt for the user for a password, and if does not work then just use the master@192.168.0.100
  + Type: ssh-copy-id slave
    - Prompt the user for a password.
  + You should be able to just type ssh slave and ssh master and not have to enter a password.
  + If ssh username does not work then you might have to setup another /etc/hosts file with the master and slave. (NOT SURE THOIUGH).
* Overview: Alright so we just setting up the keyless enetries with SSH for all the users. So the mpiusers should be able to ssh into master and the slave machine without a password. It should be able to ssh master or slave without the address on the end as well. For master and slave, they should be able to SSH into the mpiuser account, and into each other. If done correctly then we should be good to go.

**Part 7 (Checking the system):**

* It would not hurt to reboot the system, or to do a: sudo service SSH restart.
* Go to the master machine.
* Type: scp –r ~/mpich/mpich(tab)/examples –p mpiuser@master:
  + Should just move the directory to the mpiuser /home directory.
* Log into the mpiuser, you should see the examples directory in the /home directory.
* Type: mv examples ~/cloud
* Type: cd ~/cloud
* Type: sudo cp ~/examples/cpi ~/cloud.
* Type: sudo nano hostfile
  + Type: master(enter)
  + Type: slave(enter)
  + Save and close the file.
* Lets run the cluster, we should be good to go.
* Type: mpirun --hosts master,slave hostname
  + It should output master and then slave on a newline. If it does not the new have setup the system wrong.
* If that worked then lets try running another.
* Type: mpirun –np 10 --hosts master,slave ./cpi
  + Should run fine and you should be able to see the processes being ran on the slave and master machine.
* Overview: Congrats you have just setup you’re a cluster using Virtualbox and Ubuntu. There are more examples that you can use in the examples directory, you just might have to compile them.

**Part 8 (Finished! / Overview):**

* We have just successfully built a cluster; we are on our way to building a supercomputer! So there are many ways that we could’ve built a cluster, and debugging one on why it is not working is tough, especially with MPICH. So if you are having issues during the process, do not try to skip any of these processes. This was the way I did it in 2017. Sorry if there are better ways to do the previous steps, I am not a networking master or cluster Protégé. This field is always changing and settings are always being tweaked or changed.
* There will be a lot of different ways to do this online, such as installing MPICH into the cloud directory and then having to export the paths to that location, this did not work for me. I also tried installing MPICH through the wget command and there were issues down the line, so just download from the Internet, if you can’t then just use filezilla to transfer it.
* Keep snapshotting your machines if you have success, with this guide it should only take you one time. I have taken the time to layout every command that I did and my options on it. Another tip would be too add another network adapter to the master machine called Bridged Adapter. With this you can use putty from your main computer using Virtualbox and use putty to get into your server. Just a thought.

**Part 9 (Helpful commands):**

* Here are some useful commands to use in the process.
  + ssh –vvv user@address
    - This will turn on the debugger for ssh.
  + ifconfig, ifup, ifdown
  + sudo service networking restart
  + sudo /etc/init.d/network restart.
  + sudo service ssh restart
  + sudo /etc/init.d/ssh restart
* There are a ton of systemctl commands to learn but I didn’t really have to use these. Also don’t be afraid to consult your man pages on some of these commands.
* Here are some useful websites, sorry if they don’t work after some time.
  + <http://mpitutorial.com/>
  + <https://docs.oracle.com/cd/E19356-01/820-3176-10/ExecutingPrograms.html>
  + <https://stackoverflow.com/questions/26801970/mpirun-token-slots-not-supported>