BUILD YOUR OWN HIGH PERFORMANCE CLUSTER STEP BY STEP

A Study Note by ACTION lab, Depts of CEE at Mississippi State University

Dr.Pengfei(Taylor) Li   Peirong(Slade) Wang

Last updated in MAY 24th | 2018

Platform: CentOS 7

Virtual machine: Virtual box

openmpi version 2.1.1

In our example we use 10.0.1.2 and 10.0.1.3 as main node and computing node Ip address.

If you are not using virtual machine. Just manually configure your IP address on your system.

1. Set up DHCP service and configure your IP address (In Virtual box)

1. Close all the nodes
2. Open settings
3. Remove your install ISO file in storage
4. Open File Explorer on Windows. Go to C:\Program Files\Oracle\VirtualBox (where you install your virtual box). Hold Shift and Right Click white space on this window. Click Open Command Window Here. We will create an Internal Network with VirtualBox's built-in command. It will mimic our DHCP server.

   ```
   VBoxManage dhcpserver add --netname intnet --ip 10.0.1.1 --netmask
   255.255.255.0 --lowerip 10.0.1.2 --upperip 10.0.1.200 –enable
   ```

   CentOS-1: 10.0.1.2

   CentOS-2: 10.0.1.3

5. restart nodes.

   Use `nmcli d` to check you network configure

6. enps08 will be connected
   enps03 will be disconnected
   use `vi /etc/sysconfig/network-scripts/ifcfg-enp0s3`
   change on boot=yes
7. restart your network

   ```
   systemctl restart network
   ```

8. check your connection
   ping 8.8.8.8 (Google`s server, to check internet connection)
   ping 10.0.1.3 (on CentOs1, to check ethernet connection)
   ping 10.0.1.2 (on CentOs 2, to check ethernet connection)

2. Setting up NFS (Network File System) server on Machine#1

On CentOs-1 we`ll set the machine as the NFS server. We will need to install a couple of NFS libraries.

```
yum install nfs-utils nfs-utils-lib -y
```

After finish install the NFS server and libraries, we boot the NFS server by using;

```
systemctl start rpcbind nfs-server
```

```
systemctl enable rpcbind nfs-server
```

3. Setting share folder on Machine#1
   `Mkdir /nfs`   create a folder named nfs
   `vi /etc/exports` add ip address which you want to share the folder with to exports file
   e.g. `/nfs 10.0.1.3(rw,sync,no_root_squash,no_subtree_check)`
   `exportfs -a` load the/etc/exports new changes

4. Change the firewall to allow NFS and complimenting services

   ```
   firewall-cmd --permanent --zone=public --add-service=nfs
   firewall-cmd --permanent --zone=public --add-service=mountd
   firewall-cmd --permanent --zone=public --add-service=rpc-bind
   firewall-cmd –reload
   ```

   after change the firewall restart the nfs server

   ```
   systemctl restart nfs
   ```

3. Setting up NFS Client: Machine #2

1. install nfs service and libraries

```
yum install nfs-utils nfs-utils-lib -y
```

2. make a folder where the shared folder from Machine #1 will be mounted on Machine #2.

```
mkdir -p /nfs
```

3. make sure that we can access CentOS-1, the NFS Server. Make sure that the following two commands do not return any errors.

```
showmount -e 10.0.1.2
```

```
rpcinfo -p 10.0.1.2
```

```
mount 10.0.1.2:/nfs /nfs
```

With df -h, we should see that 10.0.1.2:/nfs mount has been created at the bottom. If we create any file inside /nfs, then all the machines connected can see the same file.

```
df -h
```

Now, we test that the shared folder actually works.

```
cd /nfs
```

```
touch 123.txt
```

On CentOS-1, if we cd /nfs, we will see 123.txt is inside the folder.

4.Making NFS more automatic

When you restart the two virtual machines, the NFS shared folder will not be there. We need to set a more automatic way for the NFS client to look for the NFS folder.

On the Client, we change a file called /etc/fstab.

```
vi /etc/fstab
```

We add the following line:

```
10.0.1.2:/nfs /nfs nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

Every time, we restart the client, we can re-mount the NFS shared folder by typing mount -a.

```
mount -a
```

5. Setting up SSH Keys

Create a SSH keys folder.

```
mkdir  ~/.ssh
```

generate ssh keys

```
ssh-keygen  -t rsa  -b 4096  -C your_email@example.com
```

You can press Enter to leave the next three prompts as default.

```
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]


Your identification has been saved in /Users/you/.ssh/id_rsa.


Your public key has been saved in /Users/you/.ssh/id_rsa.pub.


The key fingerprint is:
```

```
01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your_email@example.com
```

Open the ssh folder

```
cd ~/.ssh
```

copy the public key, id_rsa.pub, to authorized_keys to enable this key for access to machine #1

```
cp  id_rsa.pub authorized_keys
```

Now, we should send the private key, id_rsa, and public key, id_rsa.pub, from machine #1 to machine #2. We use a command called scp for copying files over machines.

```
scp  ~/.ssh/id_rsa ~/.ssh/id_rsa.pub root@10.0.1.3:
```

On machine #2, we have received the private key and public key. We need to make the ~/.ssh directory on machine #2.

```
mkdir  ~/.ssh
```

Now, we copy the id_rsa and id_rsa.pub to the ~/.ssh folder.

```
cp id_rsa id_rsa.pub  ~/.ssh
```

We want to copy id_rsa.pub to the authorized_keys to allow machine #1 to be able to SSH to machine #2 without a password.

```
cd ~/.ssh
cp id_rsa.pub authorized_keys
```

We should be able to ssh from machine #1 to machine #2 without a password and vice versa.

On machine #1: ssh root@10.0.1.3

On machine #2: ssh root@10.0.1.2

6. **Installing openMPI**

install gcc-5.4.0 (on both machine)

It may take serval hours to make

Install wget library (on both machine)

```
yum install wget –y
```

if you Wxget file is missing:

```
yum install gtk+-devel gtk2-devel
yum groupinstall "Development Tools"
```

download the source of openmpi from http://www.openmpi.org to nfsshre folder

```
cd /nfsshare
```

```
yum install gcc gcc-c++ gcc-fortran kernel-devel -y (on both machines)
```

```
vi ~/.bashrc
```

extract the openmpi-2.1.1.tar.gz folder

```
tar -xvf openmpi-2.1.1.tar.gz
```

We will make a directory where all the compiled binaries and libraries of mpich will go.

```
mkdir /nfsshare/<YOUR FOLDER NAME>
```

configure the settings of openmpi for installation.

```
cd /nfsshare/ openmpi-2.1.1
```

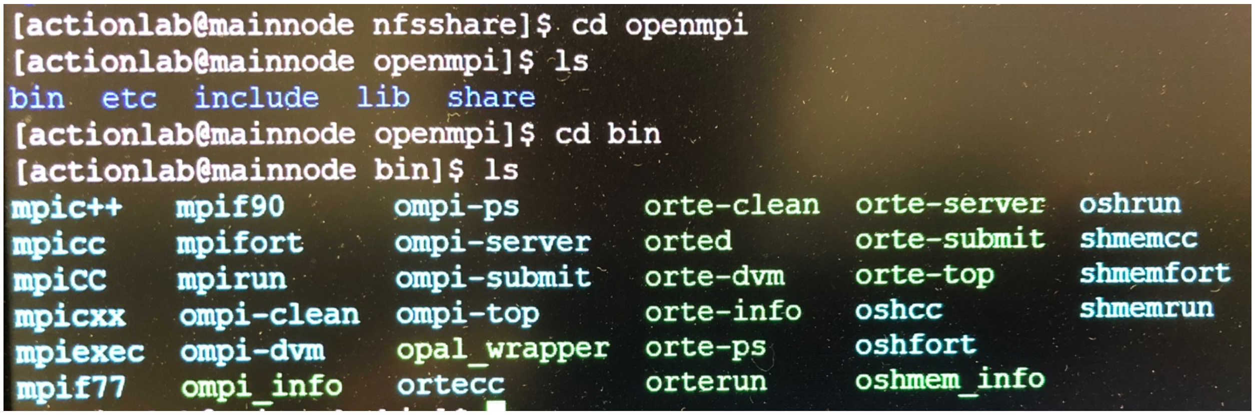
```
./configure --prefix=/nfsshare/<YOUR FOLDER NAME>
```

Install openmpi

```
make
```

```
make install
```

If we cd /nfs/<YOUR FOLDER NAME>, we will see folders containing the binaries and libraries of mpich. If we cd /nfs/<YOUR FOLDER NAME>/bin, we can see openmpi binaries like mpicc.



```
[actionlab@mainnode nfsshare]$ cd openmpi
[actionlab@mainnode openmpi]$ ls
bin  etc  include  lib  share
[actionlab@mainnode openmpi]$ cd bin
[actionlab@mainnode bin]$ ls
mpic++   mpif90      ompi-ps       orte-clean  orte-server  oshrun
mpicc    mpifort     ompi-server   orted       orte-submit  shmemcc
mpiCC    mpirun      ompi-submit   orte-dvm    orte-top     shmemfort
mpicxx   ompi-clean  ompi-top      orte-info   oshcc        shmemrun
mpiexec  ompi-dvm    opal_wrapper  orte-ps     oshfort
mpif77   ompi_info   ortecc        orterun     oshmem_info
```

Currently, we won't be able to use mpicc from anywhere on the machine. We need to change the ~/.bashrc file on machine #1 and machine #2 to globalize the mpi commands.

On both machines:

vi ~/.bashrc

At the bottom of ~/.bashrc, add the following two lines:

export PATH=/nfsshare/<YOUR FOLDER NAME>/bin:$PATH

export LD_LIBRARY_PATH="/nfsshare/<YOUR FOLDER NAME>/lib:$LD_LIBRARY_PATH"

PATH is used for bin folders, and LD_LIBRARY_PATH is used for lib folders. To reload the ~/.bashrc, type the following command on both machines:

source ~/.bashrc

7. **Using MPI binaries: Running MPI**

Go to the nfsshare folder

cd /nfsshare

create a folder for projects

mkdir /projects

create a host file contains IP address for all the IP`s that we want MPI run

vi hosts (host or other names are not working)

```
10.0.1.2 slots=1 max_slots=1

10.0.1.3 slots=1 max_slots=1
```

MPI relies on ports for TCP and UDP packet communication. We will need to stop the firewalld for the process to hop between machines.

systemctl stop firewalld (manually stop the firewall everytime when you rebooting the system)

Change your hosts names on both computers;

```
hostnamectl status
hostnamectl set-hostname
name you want to give
```

test your openmpi;

first compile your examples first:

cd /nfsshare/ openmpi-2.1.1/examples

./compile

type these commend on machine#1

mpirun –allow-run-as-root –machinefile hosts –np 2 –npernode 1 ./nfsshare/openmpi-2.1.1/examples/.a.out

8.Install Codeblocks

download codeblocks 16.01 in your nfsshare folder

extract the codeblocks-16.01.release folder

tar -xvf codeblocks-16.01.release.tar.gz

go to the codeblocks folder:

```
cd /nfsshare/codeblocks-16.01.release
```

configure and install codeblocks

```
./configure
make
make install
```

9. Appendix

videos:

1. How to set up hpc clusters on CentOS?

https://www.youtube.com/watch?v=WgUjghal_Ls&index=1&list=PLPx62H67wgD47MWNeAkvWjZURgpl6mBtu

https://www.youtube.com/watch?v=3MZcRBOsNWE&index=6&list=PLPx62H67wgD47MWNeAkvWjZURgpl6mBtu&t=1228s

2.How to install code-blocks on Linux?

https://www.youtube.com/watch?v=75UZ5ScW_TM&index=5&list=PLPx62H67wgD47MWNeAkvWjZURgpl6mBtu&t=543s

3.Change IP address static on linux.

https://www.youtube.com/watch?v=FZQBECRrpk&index=2&list=PLPx62H67wgD47MWNeAkvWjZURgpl6mBtu&t=6s

4.Install & Configure TIGER VNC Server in CentOS 7 and RHEL 7

https://www.youtube.com/watch?v=dKBAR0zUzUw&index=4&t=494s&list=PLPx62H67wgD47MWNeAkvWjZURgpl6mBtu

websites:

http://wiki.codeblocks.org/index.php/Installing_Code::Blocks_from_source_on_Linux