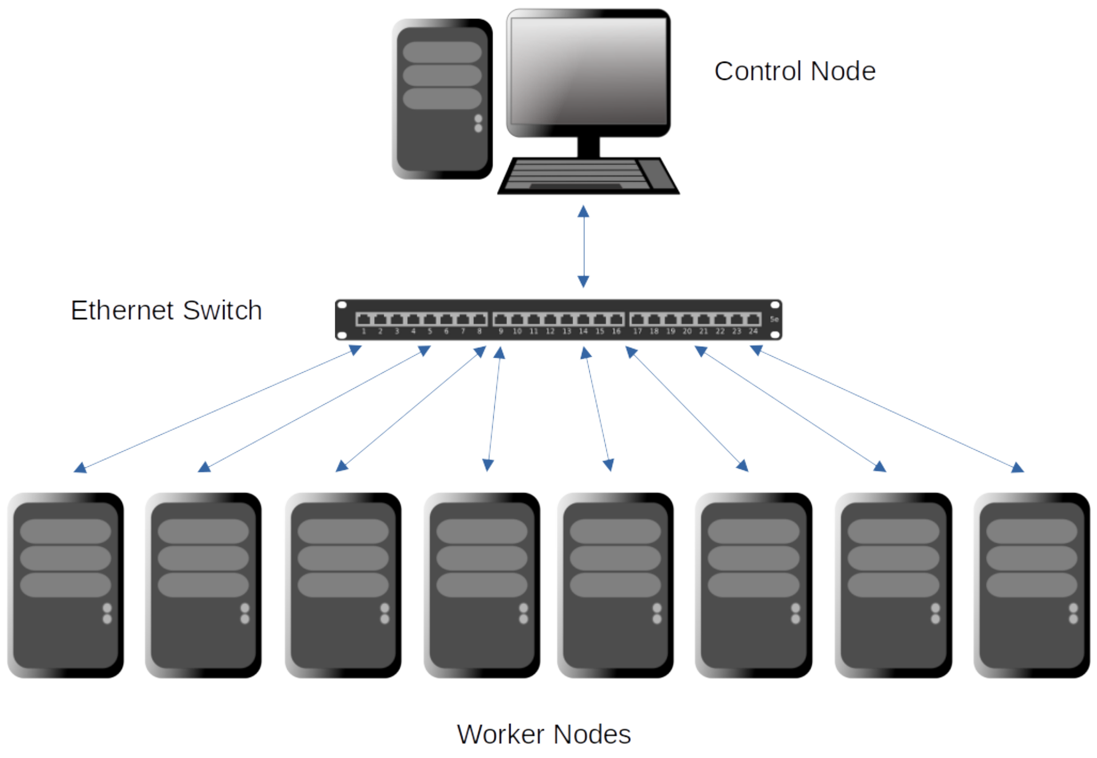
Contents

1 Introduction

The main goal of this handout is to construct a Beowulf cluster using "Commodity Off The Shelf" (COTS) hardware that will run MPI programs across the entire cluster.

> Beowulf Clusters are scalable performance clusters based on commodity hardware, on a private system network, with open source software (Linux) infrastructure. The designer can improve performance proportionally with added machines. The commodity hardware can be any of a number of mass-market, stand-alone compute nodes as simple as two networked computers each running Linux and sharing a file system or as complex as 1024 nodes with a high-speed, low-latency network.[1]

There are many different layouts for Beowulf clusters, we will concentrate on constructing a fairly simple one that consists of a single Control Node (also called a master node, manager node, head node, front node, ...) and a set of Worker Nodes (also called slave nodes, compute nodes, ...) all connected together in a local area network (LAN) with a simple Ethernet switch.



In a simple Beowulf cluster the only computer that has a monitor, keyboard, and mouse is the control node. All the worker nodes are simply the machine connected to the Ethernet switch. The control node will be able to access all the worker nodes, and use their hardware for computations, through ssh (Secure Shell) using the terminal.

As with nearly all Beowulf clusters, and supercomputers for that matter, we will be using a Linux operating system with all free open-source software (FOSS). You can use any flavor of Linux you would like

[1] `https://beowulf.org/overview/index.html`

to set up a cluster (Fedora, CentOS, OpenSUSE, Ubuntu, ...) and we will keep the discussion here fairly general except when showing actual terminal commands. Personally, I like the Debian/Ubuntu/Mint branch of the Linux distro tree, so the commands that are displayed in this document are the ones I have used to set up Mint and Ubuntu clusters, other Debian-based systems should be identical. If you choose a different distro branch you will need to alter your commands and procedures accordingly.

I have tried to be as complete as possible in these instructions but there is always the possibility that I missed something that you will encounter in setting up your system, even if you also set up a Debian-based cluster. Keep an open mind and do not get frustrated if you hit a speed-bump along the way. I hit quite a few the first time I set one of these up and it took me about 10 times as long to complete as it does for me now. Do internet searches on the errors and see what others have done to correct them. The nice thing about using Linux is that if you really mess something up you can always start over again.

2 Before You Start

2.1 About this Document & Approach

There are many ways to set up a cluster and this guide is in no way a definitive procedure. In this document I use a combination of different approaches that I have done in the past in setting up various systems in hopes that it will be quick and easy to use. You will want to read this document the entire way through before starting so that you have a feel for what will be done next. You should also read other procedures to see what the similarities and differences are so that you can make your own decisions on your particular setup. The methods below are primarily a combination of the following two web resources. I would suggest reading through these as well.

- `https://mpitutorial.com/tutorials/running-an-mpi-cluster-within-a-lan/`
- `https://www.particleincell.com/2020/ubuntu-linux-cluster/`

2.2 What You Need: Hardware

As discussed above you will need at a minimum the following hardware.

- A computer, monitor, keyboard, and mouse designated as the control node.
- A set of computers (one or more) designated as worker nodes.
- An Ethernet switch. that has enough ports for each of the control and worker nodes and one more to connect into a router or the university's Ethernet.
- Enough Cat 5e/6 Ethernet cables to connect all computers to the switch.

The computers should all be comparable in specifications. You do not want a couple fast machines and a couple slow machines on the same cluster if possible. In many cases, an MPI program may need to wait until all processes are finished before moving further so the fast machines will be sitting around waiting for the slower machines to finish. With that said, you obviously want to choose computers with the fastest clock speeds as possible (or ones with the greater number of CPU cores) but in many cases you will be constructing a cluster with older (possibly surplussed) machines. You will also want to choose the fastest Ethernet switch you can. Also with MPI programs, there is a lot of communication between the control node and the workers. If the Ethernet switch is slow then the latency in the communication will build up and hence slow the execution of the program. Again, you may be using an old or surplussed switch and not have a choice.

2.3 Configuration Decisions

If you did some research on setting up a cluster you know there are a lot of different ways to do it and a lot of different configurations, both hardware and software, that can be used. In this document we will make a couple decisions before we get started so that the final cluster will be easy to use. You are welcome to follow a different approach, but you will need to alter the procedure.

1. Choose a username that will be the same on all computers, control node and all worker nodes. In this document the user will be `mpi`, all lowercase.
2. Choose a single password what will be used for each computer/user. You will need to input this password numerous times, specifically when you install software and when you log into a worker node from the control node (during the time before you set up password-less ssh).
3. Choose computer names for each machine so that it is easy to remember which machine is which. In this document the control node computer name will be `cn` and the worker nodes will be named `wn01`, `wn02`, `wn03`, ... Some people like to label the machines and the Ethernet cables (say with masking tape) as to which machine they are and which cable goes to which machine.
4. Choose a place where you will run your MPI programs on the cluster, that is, you will create a directory that will be shared between the control node and all worker nodes. Personally, I use a directory named MPI, but you can choose any name you would like. In this document, MPI (all uppercase) will be this directory.

2.4 Setup Outline

The instructions below are a simple (and terse) outline of the essential setup for the control node, worker nodes, and their connection. In case you decide to create a non-Debian-based cluster, this will give you some points to consider in your configuration.

2.4.1 Control Node

The control node will need to have the following facilities configured.

- Linux operating system installed and updated.
- C++ installed, note that many distributions will include C but not C++.
- OpenSSH installed, server tools.
- Network file system server software installed.
- OpenMPI (or MPICH) installed, development package.
- SSH encryption key generated so that the control can log into the workers without passwords.
- Desktop software installed, text editors, IDEs, office tools, ...
- It is convenient for the control node to have two network ports. Either two wired connections or one wired and one wifi. The wired connection should have an IP address change so that the machine can control a private network (the cluster) and a public address that can access the internet.

2.4.2 Worker Node

The worker nodes will need to have the following facilities configured.

- Linux operating system installed and updated.
- The system needs to be set up for keyboard-less operation, you will probably want automatic wakes disabled, disable screen savers and power saver options.
- C++ installed, note that many distributions will include C but not C++.
- OpenSSH installed, server tools.
- Network file system software installed, the common tools are usually sufficient for the workers.
- OpenMPI (or MPICH) installed, development package.
- Change the IP address for a private network.

2.4.3 Connection

To add the worker to the cluster.

- Add the worker IP address to the hosts file of the control node. Not strictly needed but makes life a little easier.
- Add the control IP address to the hosts file of the worker node, again for convenience.
- Copy the SSH key from the control to the worker, allowing password-less access from the control to the worker.
- Add the network file system path to the fstab file on the worker.

3 Setting Up the Control Node

3.1 Procedure

1. Connect the Machine: Although fairly obvious make sure that the following is done.
    (a) Connect power, monitor, keyboard, and mouse.
    (b) Connect a cable from the control node to the switch.
    (c) Connect a cable from the switch to a port on the university's network.
2. Configure Machine BIOS Settings: On the control node there should not be much, if anything, to do here but you probably want to make sure that there is nothing strange with the settings. Always be careful when editing the BIOS of the machine, you should not do something unless you know what it will do to the operation of the machine. Most BIOS settings will have an option to reset the BIOS to factory settings, not a bad idea to do in general to start from scratch.
    (a) When the computer is booting hit the correct F-key. On Dell and Lenovo systems F2 usually takes you into the BIOS and F12 usually takes you to a one-time boot menu where going into the BIOS is an option on the menu. For other manufactures a quick internet search will tell you the correct key to press.
    (b) Normally you want to set automatic wakes to disabled.
    (c) Check that the boot sequence is the order you want it, probably the hard drive first and then maybe the flash drive (FDD on some systems). You will want to turn off any LAN boots and Windows Manager options.
    (d) I like to set the restart on fail to power off. Easier to manage when there is a power loss.
    (e) On some computers you can turn on and off the ability to boot from a USB drive, make sure that is on.
    (f) You will want to select to boot in UEFI mode and not Legacy mode, if that is even an option.
3. Install Operating System
    (a) First you will need a bootable flash drive with the operating system installer on it. This is very easy to do on a Linux system but requires administrative access which you do not have on the HPC lab machines. So you will need to do this on your own personal computer or have your instructor help you do so during an office hour. There are many freeware programs for both Windows and Mac for this. Look on the web for iso burning software or USB image writing software. In general, you will be making a bootable USB flash drive from an iso disk image.
        i. Choose the OS you want to use (obviously Linux but the flavor is up to you as long as it is FOSS). You should make sure that MPI (either OpenMPI or MPICH) is supported. I would be surprised if they were not.
        ii. Go to the download page for that OS and download the iso disk image. These are large, 2+ GB, so you will probably want a direct cable connection instead of using a wireless connection.
        iii. Use the image writing software to write the iso image to the flash drive. Obviously you will need to make sure that the flash drive is large enough to hold all the files. The iso file is compressed so the flash drive will need to be substantially larger than the iso file. A rule of thumb that usually works (an is usually a substantial overestimate) is to double the iso image size.
    (b) Once you have the bootable flash drive insert it into one of the USB ports and reboot the machine. While the machine is booting, press F12 (or appropriate key) to get the one-time boot menu. Select the flash drive from the menu, if there are both Legacy and UEFI options, use UEFI.
    (c) You will probably get another boot menu of different boot options, select the one that seems most appropriate, probably the first one.

(d) Once the computer has booted to the flash drive you are ready to install the OS. Most installers will have the option to erase the entire hard drive and install the new OS, this is what you want to do. Another approach is to open the disk utility (most installer images have a disk application that can be found under settings, options, or utilities) and delete all the partitions on the hard drive. If you do this the installer will not see any previously installed OS and default to erasing the entire drive and installing the new OS.

(e) Start the installer, in most cases it is an icon on the desktop. Most installers will ask you many questions during the install process, some things to note while installing.

 i. Keep the username for the administrative user the same as the one you chose above.

 ii. Keep the password for that user the same as the one you chose above.

 iii. Set the machine name to the one you chose above, (for example, `cn`).

 iv. Some installers will want you to set the software suites to load. Many do this by the intended purpose of the machine. You will be using this machine for development in C/C++ so you can select an option here that installs compilers and IDEs.

(f) Update the OS: Once the install process is finished you will want to reboot the machine. Most installers will have a reboot option at the end and it will tell you when to remove the installation media. Once the system is rebooted and you have logged in do the following.

 i. Go into the Driver Manager and install any hardware drivers you need. In some cases it will tell you that there are no hardware drivers to install, but if you have external cards (such as an NVidia graphics card) you will be prompted to select the driver. You will be prompted for your password to do the installation. You will need to reboot the machine before proceeding.

 ii. Open the Update Manager and install all the updates. There are always updates between the time the OS version was released and when you install it. You will be prompted for your password again. If there is Kernel update (highly likely) then you will need to reboot the machine before proceeding.

4. Install and Configure Software

(a) Install gcc and g++, that is, the C and C++ compilers. On most systems the C compiler (gcc) will be installed by default but the C++ compiler will not. If you installed a Debian-based OS, open the console/terminal and run the command,

```
sudo apt install build-essential
```

Note that the `sudo` elevates your privileges to that of a "super user". In other words, you can do anything to the system. **Be very careful when using the sudo command!** It is easy to set something or remove something that can adversely effect the running of the computer. One nice thing about running Linux is that if you really mess something up you can just start over.

(b) Install OpenSSH, this will allow the computer to access the other computers in the cluster, and vice-versa. From the terminal run,

```
sudo apt install openssh-server
```

(c) Install the network file system software. This will allow you to create a place on the control node that the worker nodes can see, needed to run MPI programs. From the terminal run,

```
sudo apt install nfs-kernel-server
```

(d) Install MPI. There are two main distributions for MPI, OpenMPI and MPICH. It is up to you which one you install, the code we write in class should compile and run fine with either. I have had some difficulties with MPICH on Debian systems so I use OpenMPI. From the terminal run,

```
sudo apt install libopenmpi-dev
```

(e) Generate an RSA Key. In a Beowulf cluster you do not want to have to log the control node into each of the worker nodes every time you start up an MPI session. It would not be all that bad with 2 or 3 worker nodes but even with 5–10 (let alone the possibility of hundreds) of worker nodes this can easily get out of hand, so you want the control node to ssh into all the worker nodes without needing a password. To do this we generate an encryption key (we will use the RSA algorithm, others are possible as well) and share the key with the workers in the cluster. This will allow password-less access to the worker node from the control node. From the terminal run,

```
ssh-keygen -t rsa -b 4096
```

and accept all default settings.

(f) Install any software you would like on the control node. It is on this node that you will be editing, running, and possibly debugging MPI code. Linux systems have a software manager that contain thousands of free software packages. You can also download and install packages that are not in the repository but most likely all you need for this class will be in the repository. Some suggestions are below, some of these may already be installed.

- GUI text editors such as Kate, Gedit, xEd, emacs, . . .
- Text based text editors such as nano, vim, emacs, . . .
- Okular document viewer (nice for PDF files)
- LibreOffice, complete office suite with word processor, spreadsheet, presentation package, image drawing system, database, and mathematical equation editor.
- Optional: Most technical documents in mathematics and computer science use a typesetter called TEX (or its derivative LATEX). This system has a bit of a learning curve to it but if you want to install it you should install texlive (probably `texlive-full`) and an editor for the language, I personally like the texmaker editor. If you install texmaker make sure you also install Okular.

5. Create the shared directory where we will be running our MPI programs. Open up the terminal and run `mkdir MPI` this will create a new directory named `MPI`. get a directory listing to confirm the creation of the directory.

6. Edit the exports file for this shared directory.

(a) Run the terminal command,

```
sudo nano /etc/exports
```

The nano is a text based editor. Since you are on the control node you could use a GUI editor and replace nano by xed. You can also use a different text based editor like vim, whatever you are more comfortable with. Add to the end of the file the following line,

```
/home/mpi/MPI *(rw,sync,no_root_squash,no_subtree_check)
```

Save the file and exit the editor.

(b) Now from the terminal run the following two commands.

```
sudo exportfs -a
sudo service nfs-kernel-server restart
```

At this point you can close the terminal window.

7. Set the IP Address. We want to make this cluster independent of the university's internet system. We will be running the cluster off-line in most cases. To do this we will change the IP address of the wired network connection to make the network private. Most distros have a GUI for network settings in their set of system settings or tools. You will want to make the following changes.

   (a) Select the iPv4 settings.
   (b) Set the IP Address to 10.0.0.1
   (c) Set the Network Mask to 255.255.255.0
   (d) Turn off the automatic DNS server, no need to put any address in this.
   (e) Set or leave the Routes to automatic.

8. Reboot

   Once the system has rebooted you will probably not be able to access the web, due to the change in IP Address and the way the university system is set up. If your control node also has wireless access you can use it to access the web, you can also use a USB to Ethernet converter (once set up) to do so as well. If none of these is an option you can change the IP on the control node back to automatic to access the web and then back to the above settings to access your LAN cluster.

3.2 Notes

If you are like me you want to test things as you go along so that you know that everything is set up correctly. Most of this setup is fairly friendly but here are a couple things you can do to see thet the software tools were installed correctly.

1. From the terminal run,

   ```
   g++
   ```

   If you get

   ```
   g++: fatal error: no input files
   compilation terminated.
   ```

   then the C++ compiler was installed.

2. From the terminal run,

   ```
   mpic++
   ```

   If you get

   ```
   g++: fatal error: no input files
   compilation terminated.
   ```

   then the MPI compiler was installed.

3. From the terminal run,

   ```
   mpirun hostname
   ```

   If you get

   ```
   cn
   cn
   cn
   cn
   cn
   cn
   cn
   cn
   ```

where `cn` is the machine name and the number of times it prints is the number of cores on your CPUs then MPI was installed completely.

4. From the terminal run,

   ```
   ip a
   ```

   You will get something like

   ```
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
       link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
       inet 127.0.0.1/8 scope host lo
          valid_lft forever preferred_lft forever
       inet6 ::1/128 scope host
          valid_lft forever preferred_lft forever
   2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
       link/ether 48:4d:7e:ee:80:e4 brd ff:ff:ff:ff:ff:ff
       inet 10.0.0.1/24 brd 10.0.0.255 scope global noprefixroute enp0s25
          valid_lft 583822sec preferred_lft 583822sec
       inet6 fe80::abcd:b18a:259c:3b7/64 scope link noprefixroute
          valid_lft forever preferred_lft forever
   ```

   If one of the lines starts with

   ```
   inet 10.0.0.1/24 brd 10.0.0.255 scope global noprefixroute ...
   ```

   Your IP address has been changed correctly.

5. From the terminal run,

   ```
   sudo systemctl status sshd
   ```

   If you get something like,

   ```
   ssh.service - OpenBSD Secure Shell server
   .
   .
   .
   ```

   Your ssh was installed correctly.

We will test the nfs install later, once we set up and link a worker node into the cluster and run an MPI program over the cluster.

4 Setting Up a Worker Node

The way we set up this process is so that the worker node can be configured separately from the control node and once placed on the network we can do a final configuration of both through the control node. Hence you can set up all the worker nodes at once and hook them to the control node or you can do one worker node at a time, whichever is more comfortable. With this process layout it is also easier to follow if you later want to add some nodes to your current cluster.

4.1 Procedure

The procedure is nearly the same as with the control node, but there are some differences.

1. Connect the Machine: Although fairly obvious make sure that the following is done.
   (a) Connect power, monitor, keyboard, and mouse.
   (b) Connect a cable from the worker node to the switch.
   (c) Connect a cable from the switch to a port on the university's network.
2. Configure Machine BIOS Settings: On the worker node there are some settings you need to be aware of.
   (a) When the computer is booting hit the correct F-key (usually F2 or F12 on Dell and Lenovo systems) to have the computer open up the boot menu or go directly into the BIOS settings.
   (b) You want to set automatic wakes to disabled, i.e. USB, LAN, ... If not then it may become difficult to shutdown the system gracefully.
   (c) Check that the boot sequence is the order you want it, probably the hard drive only.
   (d) Set the restart on fail fo power off. Easier to manage when there is a power loss.
   (e) Set the system for keyboard-less operation. Some computers will check if a keyboard is attached and halt if one is not there. Workers do not have (or need) keyboards so if there is a keyboard-less operation setting or a setting for an error on no keyboard make sure that the option is set for keyboard-less operation.
   (f) On some computers you can turn on and off the ability to boot from a USB drive, make sure that is on.
   (g) You will want to select to boot in UEFI mode and not Legacy mode, if that is even an option.
3. Install Operating System
   (a) Put the bootable flash drive into one of the USB ports and reboot the machine. Press F12 (or appropriate key) to get the boot menu. Select the flash drive from the menu, if there are both Legacy and UEFI options, use UEFI.
   (b) You will probably get another boot menu of different boot options, select the one that seems most appropriate, probably the first one.
   (c) Once the computer has booted to the flash drive you are ready to install the OS. Most installers will have the option to erase the entire hard drive and install the new OS, this is what you want to do. Another approach is to open the disk utility (most installer images have a disk application that can be found under settings, options, or utilities) and delete all the partitions on the hard drive. If you do this the installer will not see any previously installed OS and default to erasing the entire drive and installing the new OS.
   (d) Start the installer, in most cases it is an icon on the desktop. Most installers will ask you many questions during the install process, some things to note while installing.

   i. Keep the username for the administrative user the same as the one you chose above.
   ii. Keep the password for that user the same as the one you chose above.
   iii. Set the machine for automatic login. You do not want to have to log into each node on startup.
   iv. Set the machine name to the one you chose above, (for example, `wn01`, `wn02`, `wn03`, . . .).
   v. Some installers will want you to set the software suites to load. Many do this by the intended purpose of the machine. You will be using this machine for development in C/C++ so you can select an option here that installs compilers and IDEs.

(e) Update the OS: Once the install process is finished you will want to reboot the machine. Most installers will have a reboot option at the end and it will tell you when to remove the installation media. Once the system is rebooted and you have logged in do the following.

   i. Go into the Driver Manager and install any hardware drivers you need. In some cases it will tell you that there are no hardware drivers to install, but if you have external cards (such as an NVidia graphics card) you will be prompted to select the driver. You will be prompted for your password to do the installation. You will need to reboot the machine before proceeding.
   ii. Open the Update Manager and install all the updates. There are always updates between the time the OS version was released and when you install it. You will be prompted for your password again. If there is Kernel update (highly likely) then you will need to reboot the machine before proceeding.

4. Set a few system settings. Since the worker nodes are to run without user input there are a few things we should set to make this easier.

   (a) Turn off any welcome screens.
   (b) Turn off all power savers and screen savers. You do not want the system to go to sleep while you are running a program.
   (c) Set power button to shut down the computer. If all else fails you may need to use the power button to turn the machine off. Hopefully you will not need to do this but if so you don't want the machine asking you for input when there is no keyboard or monitor attached.

5. Install and Configure Software

   (a) Install gcc and g++. From the terminal run,

```
sudo apt install build-essential
```

   (b) Install OpenSSH. From the terminal run,

```
sudo apt install openssh-server
```

   (c) Install the network file system software, note that this is different than the control node. From the terminal run,

```
sudo apt install nfs-common
```

   (d) Install MPI. From the terminal run,

```
sudo apt install libopenmpi-dev
```

6. Create the MPI directory like we did on the control node. From the terminal, run the command `mkdir MPI`

7. Set the IP Address.

   (a) Select the iPv4 settings.
   (b) Set the IP Address to 10.0.0.X
   Where X is 2, 3, 4, . . . For example, worker node (`wn03`) would have an IP Address of 10.0.0.4

   (c) Set the Network Mask to 255.255.255.0
   (d) Turn off the automatic DNS server, no need to out any address in this.
   (e) Set or leave the Routes to automatic.

8. Reboot

   At this point the worker node is ready to attached to the cluster and should be able to be accessed from the control node without any input devices.

4.2 Notes

- Testing the installation here is the same as the control node tests above.
- Although it is not necessary that the workers use the same OS as the control it will probably simplify operation.
- Make sure that the same version of MPI is installed on all workers and the control. So if you installed OpenMPI on the control do the same on the workers and if you installed MPICH on the control do the same for the workers.

5 Adding the Worker Node to the Cluster

At this point we will assume that the control node is the only one with the monitor, keyboard, and mouse. All nodes are connected to the Ethernet switch and there is no connection of the Ethernet switch to the university internet. All computers have been turned on and booted successfully.

5.1 Procedure

1. At this point you can do all of the work from the control node using the terminal. I would open two terminal windows, one that is on the control node and one that will be using ssh to access the worker node. It is easier to change windows than logging off and on the workers all the time.

2. First we will set up names for the machines so that ssh is easier to use.

   (a) On the control node we are going to edit the hosts file. This takes administrator privileges so we will sudo the editor.

      i. In the terminal run,

      ```
      sudo nano /etc/hosts
      ```

      The nano is a text based editor. Since you are on the control node you could use a GUI editor and replace nano by xed. You can also use a different text based editor like vim, whatever you are more comfortable with. The file probably looks something like the following.

      ```
      127.0.0.1 localhost
      127.0.1.1 cn

      # The following lines are desirable for IPv6 capable hosts
      ::1     ip6-localhost ip6-loopback
      fe00::0 ip6-localnet
      ff00::0 ip6-mcastprefix
      ff02::1 ip6-allnodes
      ff02::2 ip6-allrouters
      ```

      Add to the bottom of the file entries for both the control node and all the worker nodes. Each line will be similar to the first two lines of the current file, each is of the form

      ```
      <IP Address>  <Name>
      ```

      So once the file is edited it will look something like,

      ```
      127.0.0.1 localhost
      127.0.1.1 cn

      # The following lines are desirable for IPv6 capable hosts
      ::1     ip6-localhost ip6-loopback
      fe00::0 ip6-localnet
      ff00::0 ip6-mcastprefix
      ff02::1 ip6-allnodes
      ff02::2 ip6-allrouters

      10.0.0.1   cn
      10.0.0.2   wn01
      10.0.0.3   wn02
      10.0.0.4   wn03
      ```

      Save the file and exit the editor. Of course, if you named your computers differently or have more then just 3 workers there would be the obvious alterations to these edits.

      ii. Now we will test the ssh installations. In the terminal you are using for the workers run the following,

```
ssh wn01
```

At this point you will be asked for the password for the worker node login. If you set up your cluster as above this is the same as the control node login password. If all is successful the commandline prompt will be something like `mpi@wn01:~`. In other words, the computer you are now on is `wn01`. Also, the control node recognizes the correspondence between the name `wn01` and its IP address `10.0.0.2`. You might also be asked (before the password) if you want to add the computer to a local list on the control node, select yes for this.

(b) Now we will do the same to the hosts file on the worker node. In the terminal where you are on the worker node run the command,

```
sudo nano /etc/hosts
```

Again you can use a different text based editor than nano, like vim. Since you are now terminal only on the worker you cannot use a GUI editor, so replacing nano with xed will not work. Since the worker node only needs to know about itself and the control node you will only add two lines to this file, no matter how many nodes are connected to the cluster. Add in the line for the control node and for the worker node. For example, if this is worker node 3 we would add the following to the end of the hosts file of worker node 3,

```
10.0.0.1   cn
10.0.0.4   wn03
```

Save the file and exit the editor.

3. Now we are going to set it up so that we can ssh from the control node to the worker node without needing a password. In the control node terminal run,

```
ssh-copy-id wn01
```

Where `wn01` is the name of the worker node you are adding. At this point the worker node can be logged into without a password. To test this go to the worker node terminal and run the `exit` command. The prompt in that terminal should now show the control node computer name. Now ssh back into the worker (`ssh wn01`) and you should not be prompted for a password and logged in automatically.

4. Now we are going to edit the fstab file on the worker so it can see the MPI directory on the control node, this is required by MPI to share data over the cluster. In the terminal window that is logged into the worker node run the command,

```
sudo nano /etc/fstab
```

Add the following line to the end of the file.

```
cn:/home/mpi/MPI  /home/mpi/MPI  nfs
```

Save the file and exit the editor. If you used a different username or shared directory name make the appropriate adjustments to the line.

5. Restart the worker node. From the terminal logged into the worker node run the command,

```
sudo shutdown -r now
```

This will restart the worker node and apply all the changes we made to the system.

5.2 Notes

Once a worker node is in the cluster you can see if it is active from the control node without ssh-ing into it. Just ping the node. From the control node terminal run the command,

```
ping wn01
```

If all is well, the terminal screen will display information about data transfers. If there is a problem then you will see an error that the "Destination Host Unreachable".

6 Testing the Cluster

On the control node you can shut the terminal window that you were using to update the worker node but you will still need the terminal that is for the control node. Make sure that all the computers in the cluster are on and have finished booting.

1. Create the following program in your favorite editor and save it to the file `mpitest.cpp` in the `MPI` directory.

```
#include <mpi.h>

int main(int argc, char **argv) {
  MPI_Init(&argc, &argv);
  int size;
  int rank;

  char name[MPI_MAX_PROCESSOR_NAME];
  int len;

  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(name, &len);

  printf("I am %d of %d on %s. \n", rank, size, name);

  MPI_Finalize();

  return 0;
}
```

2. Navigate the terminal to the MPI directory (`cd MPI`).

3. Compile the program with the MPI compiler. Specifically run the command

```
mpic++ -o mpitest mpitest.cpp
```

4. Create an "mpi hosts file" of all the computers and their respective processor cores. These can be anywhere but we will assume that this file is being stored on the control node's mpi home directory, the default one when you logged in. The file is a simple text file that contains the computer names and the number of cores each has. Each line is of the form

```
<Name> slots=<cores>
```

So for example if the control node has 8 cores and each worker has 6 the file would look like,

```
cn  slots=8
wn01  slots=6
wn02  slots=6
wn03  slots=6
wn04  slots=6
wn05  slots=6
```

Save the file to the name allhosts. The filename can be anything we will assume the name is allhosts in the commands below. If any of your CPUs are hyper-threaded you can use the hyper-thread count. That is, if you have a 4 core hyper-threaded i7 then the number of slots is 8. With that said, most of the time when I am running MPI programs, and certainly when I am doing a Linpack benchmark, I get better performance from just using the cores and not adding in the hardware threads.

Another note here is that we are using the control node as a worker as well and its CPU will be used to run the MPI programs along with the worker nodes. Some people like to set up just the worker nodes to do the work, in this case simply remove the first line of the above example.

5. Run the test program over the cluster, from the terminal that is in control node MPI directory run the command

   ```
   mpirun -hostfile ~/allhosts mpitest
   ```

   If you are including the hardware hyper-threads in your allhosts file you will need to add in the `-use-hwthread-cpus` option.

   ```
   mpirun -use-hwthread-cpus -hostfile ~/allhosts mpitest
   ```

   The output should look something like the output below. There should be an entry for each core on each machine.

   ```
   I am 6 of 38 on cn.
   I am 7 of 38 on cn.
   I am 0 of 38 on cn.
   I am 1 of 38 on cn.
   I am 2 of 38 on cn.
   I am 3 of 38 on cn.
   I am 5 of 38 on cn.
   I am 4 of 38 on cn.
   I am 28 of 38 on wn04.
   I am 17 of 38 on wn02.
   I am 18 of 38 on wn02.
   I am 9 of 38 on wn01.
   I am 10 of 38 on wn01.
   I am 29 of 38 on wn04.
   I am 32 of 38 on wn05.
   I am 19 of 38 on wn02.
   I am 30 of 38 on wn04.
   I am 33 of 38 on wn05.
   I am 14 of 38 on wn02.
   I am 11 of 38 on wn01.
   I am 26 of 38 on wn04.
   I am 34 of 38 on wn05.
   I am 16 of 38 on wn02.
   I am 12 of 38 on wn01.
   I am 27 of 38 on wn04.
   I am 35 of 38 on wn05.
   I am 13 of 38 on wn01.
   I am 37 of 38 on wn05.
   I am 15 of 38 on wn02.
   I am 8 of 38 on wn01.
   I am 31 of 38 on wn04.
   I am 36 of 38 on wn05.
   I am 21 of 38 on wn03.
   I am 22 of 38 on wn03.
   I am 23 of 38 on wn03.
   I am 24 of 38 on wn03.
   I am 25 of 38 on wn03.
   I am 20 of 38 on wn03.
   ```

7 Final Thoughts

At this point you have a Beowulf COTS cluster ready to run MPI programs. Congratulations, the majority of computer scientists in the field have never done this. In fact, a lot of scientists that work on supercomputers have never gotten their hands dirty with the hardware or its configuration, that is the work of the vendor of the machine.

There is far more we could do here if we wanted and you are welcome to experiment with your cluster. For example, we have not installed any management software like Kubernetes, configuration management tools like Ansible or SaltStack, or workload management tools like Slurm. For a small cluster like this using the terminal and ssh is sufficient and not too cumbersome, but if you build a larger cluster you would want to look into these tools. For smaller systems a couple bash or python scripts could do your needed work.

For example, the way we set this cluster up, to turn the cluster on we simply press all the power buttons and wait for all the lights on the switch to come on and we are ready to go. Shutting down the cluster is a different story. Yes we could hold the power buttons in or unplug the cluster but as you know that would not be a “graceful” shutdown. As we have it set up we would need to ssh into each worker node one at a time, run the command `sudo shutdown now` (or `sudo shutdown -r now` to restart) which would require us to type in the password each time. We could automate this by doing a one-time edit on each worker and then running a script from the control node.

The edit on each worker is the following. From the control node terminal, ssh into the worker. Run the command,

```
sudo visudo
```

At the end of the file include the line

```
mpi ALL = NOPASSWD: /sbin/shutdown
```

Save the file and exit the editor. Again the `mpi` is the username for the computer and can be changed accordingly. What this command does is tell the commandline shutdown not to ask for a password. If you are familiar with writing bash scripts the following script will shutdown a 5 worker node cluster and leave the control node on.

```bash
#!/bin/bash

ssh wn01 sudo shutdown now
ssh wn02 sudo shutdown now
ssh wn03 sudo shutdown now
ssh wn04 sudo shutdown now
ssh wn05 sudo shutdown now
```

Similarly, the following script will restart the cluster.

```bash
#!/bin/bash

ssh wn01 sudo shutdown -r now
ssh wn02 sudo shutdown -r now
ssh wn03 sudo shutdown -r now
ssh wn04 sudo shutdown -r now
ssh wn05 sudo shutdown -r now
```

If you are more comfortable with Python the following will shutdown the same cluster.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os

os.system("ssh wn01 sudo shutdown now")
os.system("ssh wn02 sudo shutdown now")
os.system("ssh wn03 sudo shutdown now")
os.system("ssh wn04 sudo shutdown now")
os.system("ssh wn05 sudo shutdown now")
```