

Creating a Raspberry Pi Cluster: A Tutorial for Students

Written by Dr. Suzanne J. Matthews, United States Military Academy

Adapted from [instructions](#) provided by Dr. Simon Cox, University of Southampton

ABOUT OUR RASPBERRY PI CLUSTER:

The goal of this tutorial is to create a simple Beowulf cluster using Raspberry Pis. After completing these instructions, you should have a simple Raspberry Pi Cluster with N nodes. In the examples that follow, $N=4$. However, the tutorial can be used to create clusters of any N .

One of the most common parallel patterns is the Master-Worker pattern. In this pattern, a master process (or node) is responsible for creating a set of tasks and delegating them to some $N-1$ workers. All N nodes then complete the work and report the final results back to the master.

In the case of our Raspberry Pi cluster, communication will be taken care of using the MPI library. All of our work will be done on the master node; we will write and execute our MPI programs there. Our master node will then use the MPI library to communicate with the other nodes in our cluster, and complete the work in parallel. This is the first part in a series of labs exploring our Raspberry Pi cluster.

Are you ready? Let's get started!

THINGS YOU WILL NEED BEFORE-HAND:

- A laptop (with SD card reader)
- $N=4$ raspberry pis
- $N=4$ raspberry pis w/ power supply, cat5, and sd card
- a switch with access to the internet (or a router)
- 3-d printed cases parts (for each board: 2 base boards, 4 connectors)



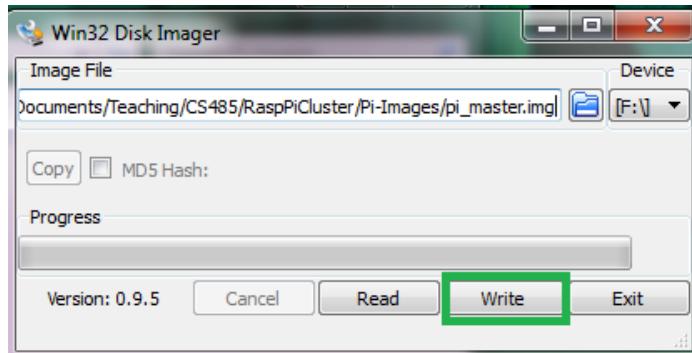
This is what each Raspberry Pi should look like when assembled!

Begin by assembling only two of your Pis. Keep the SD cards and cat5s aside for now.

SETTING UP THE MASTER NODE:

The SD card that comes with our Raspberry Pi is used to house its *operating system*. To set up our master node, we must first install an image of our desired operating system on it. In essence, an image allows us to get a set of desired software onto our system quickly and relatively painlessly. We provide an image for you, called pi-master.img that you can download from the course website.

1. Install [Win32 Disk Imager](#) (you may need to use admin privileges).
2. Download the pi-master.img file from the course website.
3. Insert an SD card into SD card reader slot.
4. Using Win32-disk-imager, select the drive the SD card is in (usually the F drive). Be absolutely sure that the selected drive is the one associated with the SD card!
5. Use the folder icon to navigate to where you downloaded the image. Click "write". This process may take a few minutes.



6. Once the process is done, safely eject the SD card and boot up your Raspberry Pi. The login is "pi" and the password is "student". While we will do the majority of the work on the command line, you can gain a more typical "desktop" view by typing in the command `startx`.
7. Use the cat5 cable to connect your Raspberry Pi to your router. Type in `ifconfig` to get your IP address. You should see something like 192.168.1.123. The last 3 numbers will likely be different on your machine. It should NOT be 255!
8. Replace the IP address in the provided `machine_file` file in the `mpi_test` directory by using the following command (replace 192.168.1.123 with the IP address you found in the previous step). Next, `cd` into the `mpi_test` directory.

```
echo '192.168.1.123' > ~/mpi_test/machine_file
cd ~/mpi_test/
```

9. Ensure that MPI works by typing in the following command:

```
mpiexec -f machine_file -n 2 ~/mpich_build/examples/cpi
```

You should get the following:

```
Process 0 of 2 is on raspberrypi  
Process 1 of 2 is on raspberrypi  
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
```

If you make it this far, great job! You are ready to create worker nodes.

SETTING UP THE WORKER NODES – PART I

Our master node will “communicate” with our worker nodes using the Secure Shell (SSH) protocol. In this next step, we need to give our master node the proper login credentials so it can seamlessly connect to the worker nodes as needed. To accomplish this, we will add ssh credentials to our master node, and test our setup using a new worker node. All the commands below (unless otherwise noted) are typed on the master node.

1. Create an image for your new worker node by repeating steps 3-6 in the previous section. Connect the new worker node to the router and boot it up.
2. On the master node, generate an ssh key by typing in the following:

```
cd  
ssh-keygen -t rsa -C 'pi@raspberrypi'
```

Use the default location. Do NOT enter a passphrase!

3. Determine the IP address of the worker node. Usually with routers it is one number above the IP of your current node. You can test this by typing:

```
ping 192.168.1.124
```

If a route to the machine exists, you will get a response. Be sure to check your connections to the router to ensure that the master node is plugged into the “1” location, and the worker node is plugged into the “2” location. In the examples that follow, 192.168.1.124 is the IP of the worker node. Please be sure to replace this with the IP address of your worker node!

4. Copy the log-in credentials to your worker node by typing in the following:

```
cat ~/.ssh/id_rsa.pub | ssh pi@192.168.1.124 'cat >> .ssh/authorized_keys'
```

Where 192.168.1.124 should be replaced with the IP of the worker node.

5. When this is done, ensure you can ssh into the worker node without the need for any credentials by typing in the following:

```
ssh pi@192.168.1.124
```

You should automatically be logged on to the machine, without being asked for any passphrases or passwords. If this doesn’t happen, you did something wrong, and will need to repeat steps 2-4.

6. Let's change the hostname for this machine. Type in the following command

```
sudo nano /etc/hostname
```

to launch the nano editor. Replace what's there with something like worker001. Restart the machine to see changes:

```
sudo shutdown -r now
```

This will restart the worker node, and kick you out of the ssh session.

7. Next, let's repeat the test from the previous section. `cd` into the `mpi_test` folder and edit the file `machine_file` and add the IP of your new worker node:

```
echo '192.168.1.124' >> ~/mpi_test/machine_file  
cd ~/mpi_test/
```

Rerun the MPI test:

```
mpixexec -f machine_file -n 2 ~/mpich_build/examples/cpi
```

You should get something like the following:

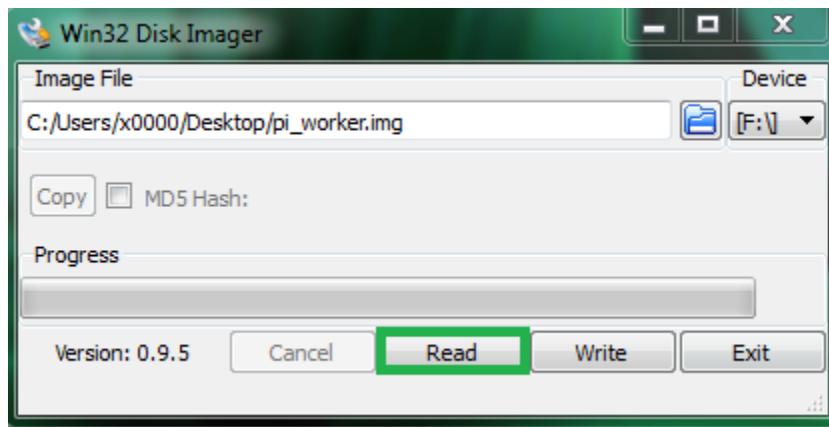
```
Process 0 of 2 is on raspberrypi  
Process 1 of 2 is on worker001  
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
```

If the above works, great job! You are ready to create additional worker nodes! Unlike our test from the previous section, we are now running our test on multiple nodes!

SETTING UP THE WORKER NODES – PART 2

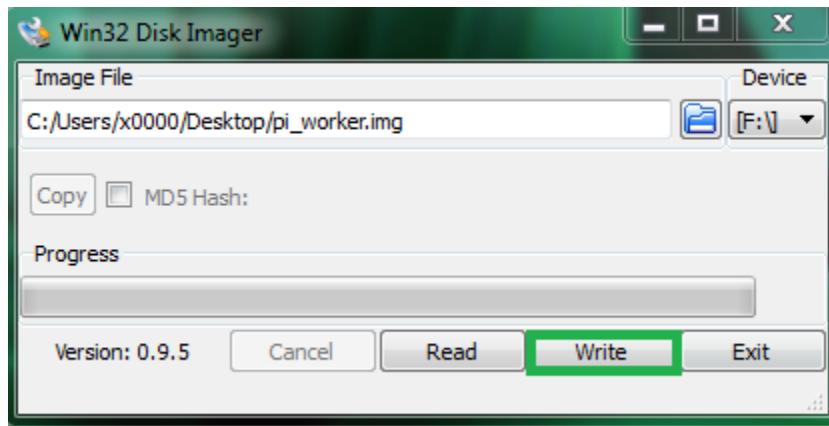
Our worker node now contains the proper SSH credentials our master node needs to log-in seamlessly. To allow our master node to be able to log in to other nodes, we must copy the image of the worker node we just created onto the other worker nodes we plan to create.

1. Place the SD card of the worker node you created in the previous section into your laptop. Fire up Win32-disk-imager.
2. Type a path into the text box:
`C:\Users\x0000\Desktop\workerNode.img`
where x0000 is your xNumber. Select the drive that contains the SD card.
Click the “read” button. This will copy the image that is on your SD card to your desktop.



Once the process is done, remove the SD card, and place it back into your worker node.

3. Insert a new SD card. In Win32-disk-imager, select the image you just wrote to your desktop. Ensure the SD card's drive is selected in the window. Click "write". This process may again take a while.



4. Repeat the previous step for each additional worker node.
5. Once you are done, load the new SD cards into your raspberry pis, and connect them to your router. Figure out the ips for each, ssh in, and change their hostnames using the process in step 6 in the previous section. You want each worker node to have distinct names. So, if you are creating two new worker nodes, you may want to name them `worker002` and `worker003`. Remember to poweroff the worker nodes in order to see the reflected changes!
6. Repeat the MPI test we did in the last two sections. `cd` into the `mpi_test` folder, and edit `machine_file` to include the IP addresses of the new worker nodes. Then, re-run the test:

```
7. echo '192.168.1.125' >> ~/mpi_test/machine_file
   cd ~/mpi_test/
   mpiexec -f machine_file -n 2 ~/mpich_build/examples/cpi
```

You should get something like the following:

```
Process 0 of 4 is on raspberrypi
Process 1 of 4 is on worker001
Process 3 of 4 is on worker003
Process 2 of 4 is on worker002
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
```

Remember these can be out of order (so don't worry about that). If everything above works, hooray! You are done!

TROUBLESHOOTING

If the above test fails:

- Check and make sure that all the ip addresses in your machine_file are valid. If not, fix your machine_file with the appropriate ids.
- Ensure that you can ssh into the worker nodes from the master nodes without needing a password. If you are asked to provide a password, that means the host's ssh keys were not added to the worker's set of authorized keys. Follow step 4 in the "Setting up worker nodes" section. Reboot the worker node!
- Ensure that you can ssh from each worker node into the master node (this is an issue with some routers – does not appear to be a problem when connecting to a switch):
 - Edit /etc/hosts.allow and add the following line:

```
sshd: ALL
```

- If you CAN ssh from each worker node into the master node, BUT require a password to do so, you may need to add generate an ssh key for each worker node and add it to the master's list of known hosts. Replace worker001 with the name of each respective host, and the IP below with the master's IP address:

```
cd
ssh-keygen -t rsa -C 'pi@worker001'
cat ~/.ssh/id_rsa.pub | ssh pi@192.168.1.123 "cat >> .ssh/authorized_keys"
```

Be sure to restart the master node when you are done.

BUILDING THE CASE

Now we have N raspberry pi's networked together. Place boards in a 2 x 2 configuration. Place 2 middle connectors to connect the 2 base boards on each layer. Place end connectors in the holes on the end. Repeat with next layer, and finish by placing remaining 2 base boards on top. If you want to generate more space between the clusters, simply add middle connectors to each layer.

Finished cluster is shown below:

