

HOW TO BUILD A RASPBERRYPI CLUSTER

SIMONE ATZENI & MOHAMMED AL-MAHFOUDH

CONTENTS

1	Introduction	2
2	Raspbian Debian Wheezy OS: Installation	2
3	Raspbian Debian Wheezy OS: Configuration	3
3.1	Server Side (main node Pi-Srv)	5
3.2	Client Side (all nodes from Pi1 to Pi8)	5
4	Concluding remarks	7

LIST OF FIGURES

Figure 1	The complete 9 nodes cluster	2
Figure 3	Network diagram	3

LIST OF TABLES

Table 1	List of parts, websites, and prices for each part and total price	2
Table 2	Configuration of each node's IP address	4

1 INTRODUCTION

RaspberryPi was invented as educational device to inspire children to study computer science. Nevertheless, in a couple of years it has become the toy of computer scientists, engineers and the curious of every age.

This project is our successful attempt to build an educational RaspberryPi Cluster composed of one main node and eight computational nodes. In this post we try to explain step-by-step our work in order to allow people to build their own cluster(s).

First of all, we need to make a detailed list of all parts we have to put together to build our cluster. Listed in the table below are all of the pieces and respective websites from which you can buy. [Amazon](#) and [eBay](#) are the most popular e-commerce websites where all these parts can be found easily and for low prices. The prices and the links indicated in the table are the information at the time we built the cluster. At the moment you are reading this post could be out-of-date. Use the name of the part to look for more updated information on Internet.



Figure 1: The complete 9 nodes cluster

Part	Qty	Price	Total
Raspberry Pi Model B	8x	\$37.50	\$300
CanaKit Raspberry Pi (512 MB) Complete Starter Kit	1x	\$59.99	\$59.99
CablesOnline 10-PACK 6inch CAT5e UTP Ethernet	1x	\$13.79	\$13.79
AmazonBasics 7 Port USB 2.0 Hub	2x	\$18.99	\$37.98
Sony 16GB SDHC/SDXC Memory Card	9x	\$12.95	\$116.55
NETGEAR 8-Port Gigabit Ethernet Switch	1x	\$41.99	\$41.99
BUFFALO AirStation N450 Gigabit Wireless Router	1x	\$83.99	\$83.99
Stacked 3 Floors Clear Raspberry PI case	3x	\$13.99	\$41.97
Total Price			\$696.26

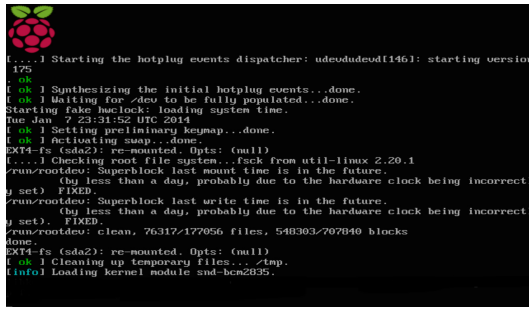
Table 1: List of parts, websites, and prices for each part and total price

2 RASPBIAN DEBIAN WHEEZY OS: INSTALLATION

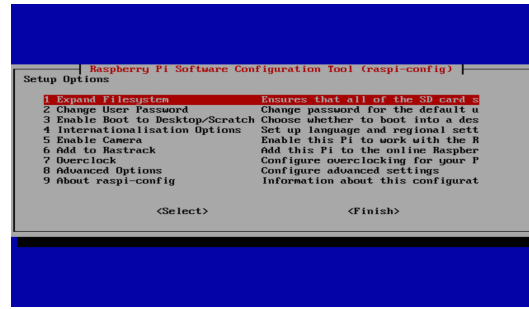
There are many Operating Systems to use with the Raspberry Pi. They are usually based on known Linux distribution like Debian, Fedora, etc. For our project we opted for a Debian-based version called [Raspbian Debian Wheezy](#). A ready-to-use image of the OS can be downloaded from [Raspbian Debian Wheezy official website](#). Once the OS image has been downloaded, it needs to be installed in the SD card. This operation should be done for all SD cards that we have, so for the number of nodes of our cluster, nine in our case. However, since we would need to configure different aspects of the OS, and in order to speed up all of these operations, we will use a freely available disk imaging utility (from command line). Let us start installing our image only on one SD card. To do this operation we will need an SD card reader. After inserting the SD card in the reader, the command to write the OS in our SD card using a Linux system is the following ¹:

```
sudo dd bs=4M if=2014-01-07-wheezy-raspbian.img of=/dev/your_device_file
```

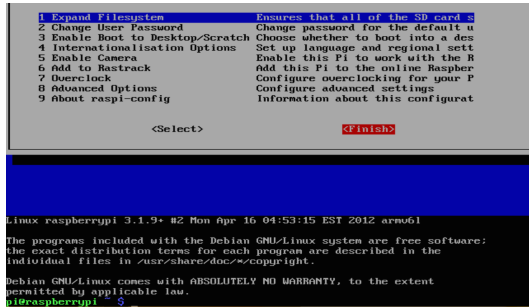
¹ Note that the “dd” command is very powerful and could overwrite any partition of our system. We must be sure that we are selecting the right device. For more detailed instructions refer to the Raspberry website. There are guides to perform this operation with different systems (Linux, Windows and Mac OS).



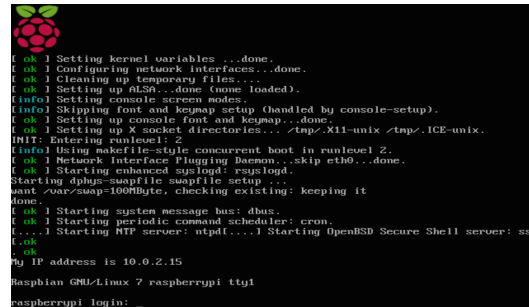
(a) Booting phase



(b) First boot options menu



(c) Filesystem expansion



(d) Login screen

We can put the SD card into the slot of our Raspberry Pi, connect keyboard, mouse and a monitor or a TV with an HDMI cable and turn it on connecting the Micro USB power supply. The first time that we boot our Raspbian operating system, we will be presented a configuration menu. We will choose “Expand Filesystem” to ensure that all the SD card storage is available to the OS. Below the screenshots of the different phases. Once the filesystem has been expanded we can login in the system with username “pi” and password “raspberrypi”.

3 RASPBIAN DEBIAN WHEEZY OS: CONFIGURATION

Now, we need to configure the network settings of the operating system. Our cluster will have the network structure shown in the picture.

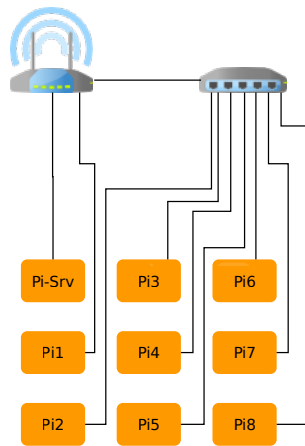


Figure 3: Network diagram

We use the **NETGEAR 8-Port Gigabit Ethernet Switch** to connect seven of the nine nodes, the remaining port is used to connect the switch to the **BUFFALO AirStation N450 Gigabit Wireless Router** and the other two nodes are also connected to the router. In our project we use a wireless router so that we can have a “mobile” cluster for educational purposes. You can modify the network architecture to better fit your needs.

As we can see from the network diagram, there is a main node called “Pi-Srv”, it will be the head of the cluster to submit the jobs on the other eight computational nodes. Furthermore, the main node will share its own filesystem to the other nodes. This way, there will be only a single control point (the “Pi-Srv”) to compile and run the programs. We start the configuration assigning a hostname and a static IP address to each node. The table summarizes the network information for each node.

Hostname	IP Address	Net Mask
raspberrypi-srv	192.168.0.80	255.255.255.0
raspberrypi81	192.168.0.81	255.255.255.0
raspberrypi82	192.168.0.82	255.255.255.0
raspberrypi83	192.168.0.83	255.255.255.0
raspberrypi84	192.168.0.84	255.255.255.0
raspberrypi85	192.168.0.85	255.255.255.0
raspberrypi86	192.168.0.86	255.255.255.0
raspberrypi87	192.168.0.87	255.255.255.0
raspberrypi88	192.168.0.88	255.255.255.0

Table 2: Configuration of each node's IP address

The information about address and mask must be inserted in the configuration file “/etc/network/interfaces” for each node, while the hostname will be inserted in the configuration file “/etc/hostname”. We also edit the file “/etc/hosts” to give each node information about the hostname of the other nodes. Now we are configuring the head node, so our configuration files will look like shown below.

/etc/network/interfaces

```

1 auto lo
2 iface lo inet loopback
3 auto eth0
4 iface eth0 inet static
5 address 192.168.0.80
6 network 192.168.0.0
7 netmask 255.255.255.0
8 broadcast 192.168.0.255
9 gateway 192.168.0.1
10 allow-hotplug wlan0
11 iface wlan0 inet manual
12 wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
13 iface default inet dhcp

```

/etc/hostname

```

1 raspberrypi-srv

```

/etc/hosts it will be the same in each node

```

1 127.0.0.1 localhost
2 ::1 localhost ip6-localhost ip6-loopback
3 fe00::0 ip6-localnet
4 ff00::0 ip6-mcastprefix
5 ff02::1 ip6-allnodes
6 ff02::2 ip6-allrouters
7 192.168.0.80 raspberrypi-srv
8 192.168.0.81 raspberrypi81
9 192.168.0.82 raspberrypi82
10 192.168.0.83 raspberrypi83
11 192.168.0.84 raspberrypi84
12 192.168.0.85 raspberrypi85
13 192.168.0.86 raspberrypi86
14 192.168.0.87 raspberrypi87
15 192.168.0.88 raspberrypi88

```

Now that we have configured the general aspects of one node we can quickly copy the entire OS (already configured) in the remaining eight SD cards. In order to do this let us power off the main node with the following command:

```
1 poweroff # when we reach the "webmin" admin console, we will clarify more about
           shutting down the whole cluster
```

Remove the card from the RaspberryPi and insert it again in the SD card reader to make a copy with the command:

```
1 sudo dd bs=4M if=/dev/your_device of=2014-01-07-wheezy-raspbian_configured.img
```

We can copy our OS image inside all the other SD cards running the following command for each card:

```
1 sudo dd bs=4M if=2014-01-07-wheezy-raspbian_configured.img of=/dev/your_device
```

At this point we have the same OS and configuration for each one of the nine RaspberryPi. The only operation that we need to do is to change the IP address and hostname according to the table. We can follow the same steps we did for the configuration of the main node. Once this phase is complete we have all nine nodes ready to take part to the cluster.

In order to assemble the cluster, let us follow the diagram shown previously and the pictures below.

As we said previously, the master node will share its own home folder to the other computational nodes. In order to do this operation let us follow the following steps:

3.1 Server Side (main node Pi-Srv)

- Installing the NFS Kernel Server package

```
1 sudo apt-get install rpcbind nfs-kernel-server
```

- Add the following line in the configuration file “/etc/exports”:

```
1 /home/pi 192.168.0.0/24(rw, sync, no_subtree_check)
```

- Restart the NFS Kernel Server

```
1 sudo service nfs-kernel-server restart
```

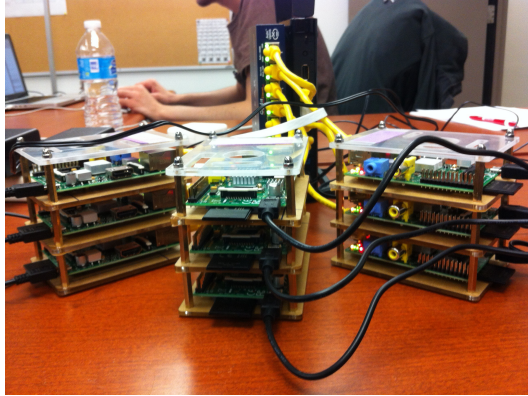
3.2 Client Side (all nodes from Pi1 to Pi8)

- Installing the NFS package

```
1 sudo apt-get install rpcbind nfs-common
```

- Add the following line in each node’s configuration file “/etc/fstab”:

```
1 raspberrypi-srv:/home/pi /home/pi nfs defaults o o
```



(a) Close-up of the completed setup



(b) The cluster completely setup



(c) Installing the heat sinks



(d) A wires mess, this is also an aspect of cluster computing that we won't have the room to discuss (cables management)

4 CONCLUDING REMARKS

By now, any thing you place in the home directory in the head node, should be reflected in each other node's home folder as if you have copied it all over to all other nodes too. Except that it will use some space from the head node only and then will be accessed in a shared manner from all other nodes. The catch to this is that if you have too many transfers and/or too much data flowing back and/or forth between the head node and the other nodes, this will slow your computation. So, make sure to reduce that traffic by placing the needed data on the actual SD card and leaving that shared NFS volume for just deploying programs.

As for the configuration of our network topology of the cluster, there must be more efficient configuration targetted at your computation. Our setup's main goal was to reduce hardware needed for connecting the cluster's nodes. Yours may differ, you may want best node-to-node communication latency in which case you may opt for a star-schema for the interconnections (i.e. a switch connecting all nodes to each other point-to-point wise). This subject goes all the way to networking and all the details of backbones and interconnect reaching to proprietary interconnects. As such, we conclude our cluster setup guide and leave the details for the curious about networking to take a complete networking course.