

# Random thoughts on coding and technology

Thoughts about Delphi, C#, .NET and Software Architecture.



HOME

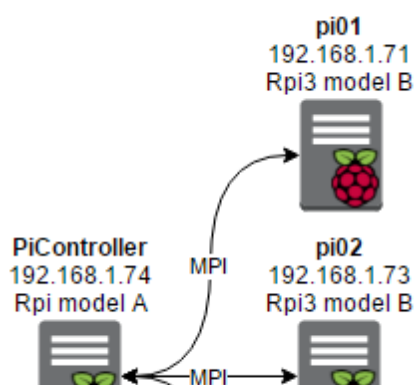
OPEN SOURCE ▾

FLICKR ANALYTICS ▾

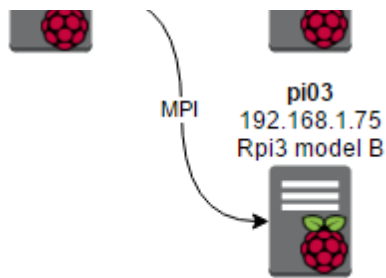
MVP ▾

## Creating a Raspberry Pi 3 Cluster - "Supercomputer", for parallel computing.

📅 22:53:00 👤 JORDI CORBILLA 📁 ARCHITECTURE, COMPUTING, LINUX, NETWORK,  
PARALLEL, RASPBERRY PI, RPI3, SUPERCOMPUTER 🔍 60 COMMENTS



In this quick article I will show you how to create your own **Raspberry Pi cluster** for [parallel computing](#) via [MPI](#) (Messaging Passing Interface) library. This is a nice summer project now that I'm free from my Master's duties until September and I have been wanting to build this for a while [^](#)  
Thanks to the low prices of the Raspberry Pi we



are now able to build this without spending too much. See below for the list of items you will need and price for the whole kit with 4 Pi's.

The main decision behind this architecture is to choose which **operating system** and **programming language** to use to implement **parallel computing**. Because of my experience with [HPC](#) (High Performance Computing) and [SGE](#) (Sun Grid Engine) the best way to achieve this is by using either [OpenMPI](#) or [MPICH3](#). These two are free open distributions, portable and very popular. As per the programming language, we have several alternatives: we could use *c++*, *c#*, *python*, etc. I could use [winIoT](#) for Rpi or simply a Linux distro. I'm a geek so I went for the latter as I do like interacting with command line interfaces, there is some beauty there that I can't explain :).

So my decision is to use a Linux distribution as OS. In this case I'm choosing [Raspbian Jessie](#) which comes with some goodies installed by default and it will allow me to install all the components I need for my little project.

The second decision to make is to choose the programming language. In this case I'm choosing **Python** as I'm very familiar with it, it has plenty of libraries available and a nice integration with MPI via [mpi4py](#) library.

The other factor to take into account here is that I have two different models of RPi and I need to make sure that whatever I install on those will work well for both instances. I won't be able to install WinIoT to my old **Rpi model A**.

## Building the cluster of Rpi's

The material that you will need is listed below with links included:

- 4 x [Rpi 3 model B](#) = 4 x £30 = £120
- 4 x [16Gb microSD card \(Kingston\)](#) = 4 x £4.84 = £19.36
- 4 x [USB to Micro USB Cable 0.5m](#) = 4 x £0.88 = £3.5
- 2 x [Multi-Pi Stackable Raspberry Pi Case](#) = 2 x £13 = £26
- 1 x [5 port desktop switch](#) = 1 x £6.49 = £6.49
- 5 x [Ethernet patch cable 0.3m](#) = 5 x £2.90 = £14.5



$$1 \times \text{USB Hub} = 1 \times £2.53 = £2.53$$

**Total = £192.38** (without considering delivery)

\*This is a common configuration but you can start with just 2 or 3 RPi's and keep adding hardware later on.

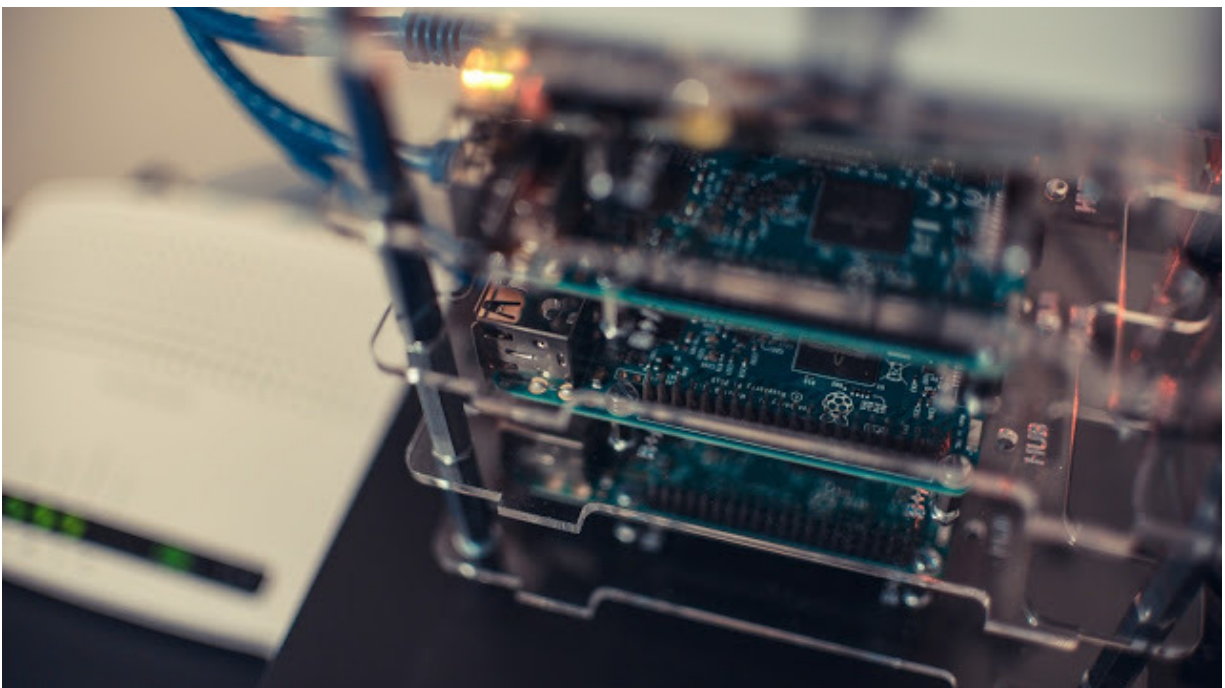


Once all the components are assembled using the stackable case you should have something like the image below:





Below the image of my cluster up and running (see configuration section for more):

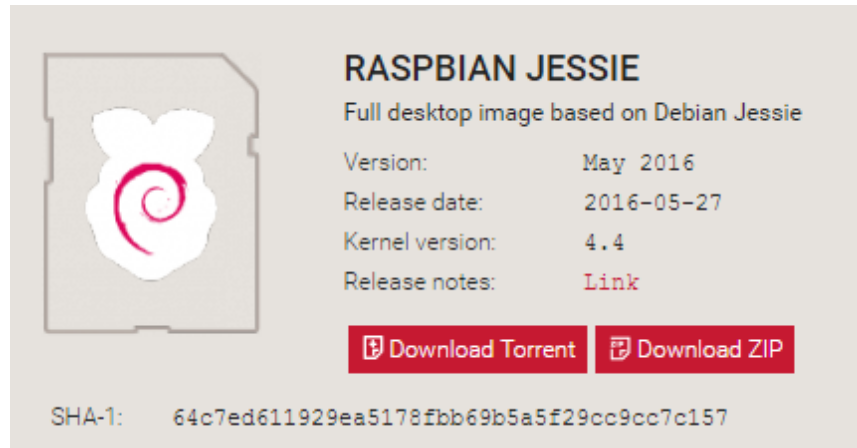


## Configuring your cluster of RPi's

The idea is to configure one of the RPi's and then just clone the SD card and plug it to the next Rpi. Here you'll find a summary description of the steps to do to get you up and running: [^](#)

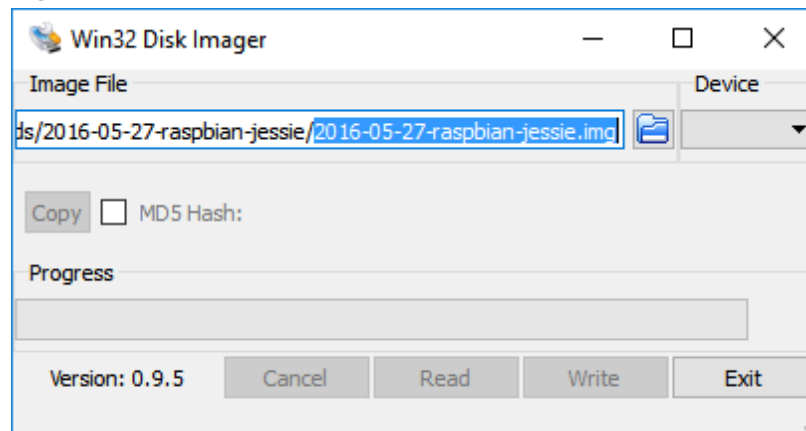
## Installing the OS

- Download [Raspbian Jessie image](#). I had some trouble downloading the zip file so I used the torrent link instead. See the version used below (4.4)



- Download [Win32DiskImager installer](#). We will use this to burn Raspbian image to our SD card.
- Download [PuTTY](#) SSH client to connect to our Rpi's.

Once the OS image is downloaded, burn it to the SD card using **Win32DiskImager**:

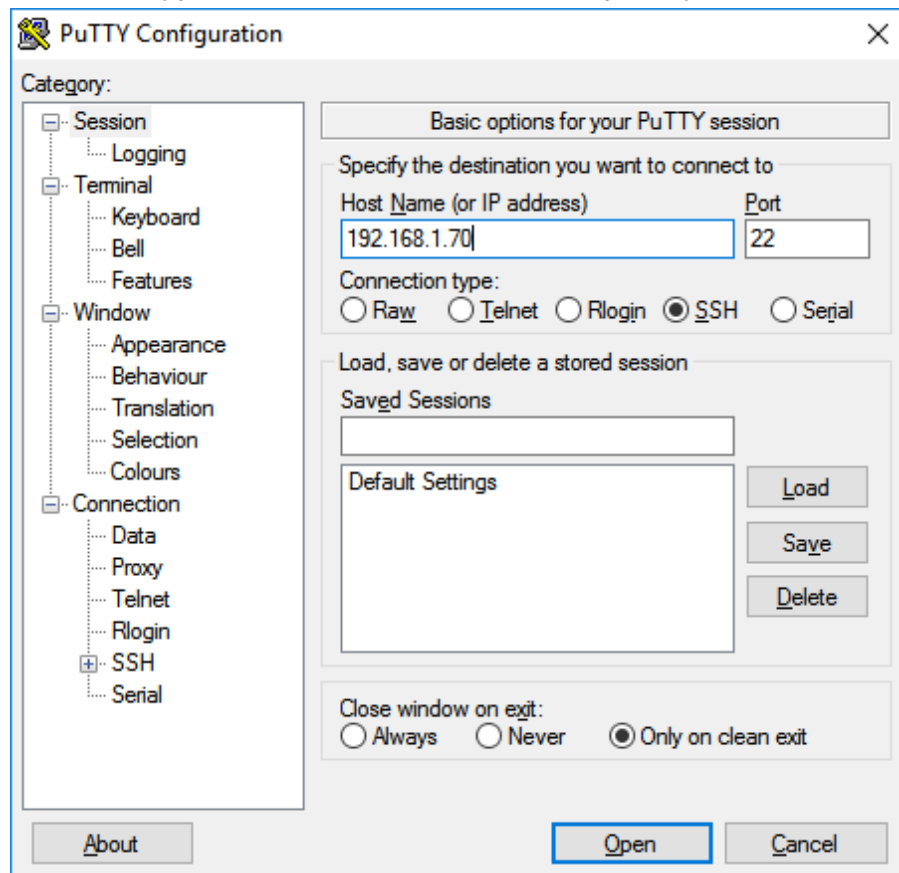


Plug the microSD card to the first Pi (my PiController in my case) and power it up. Plug the Ethernet cable and head back to your computer to access the Pi remotely.

Open a **command prompt** (I'm using Win10 as my main computer) and type "**ping raspberrypi**". By default the Rpi's are named raspberrypi so they are eas

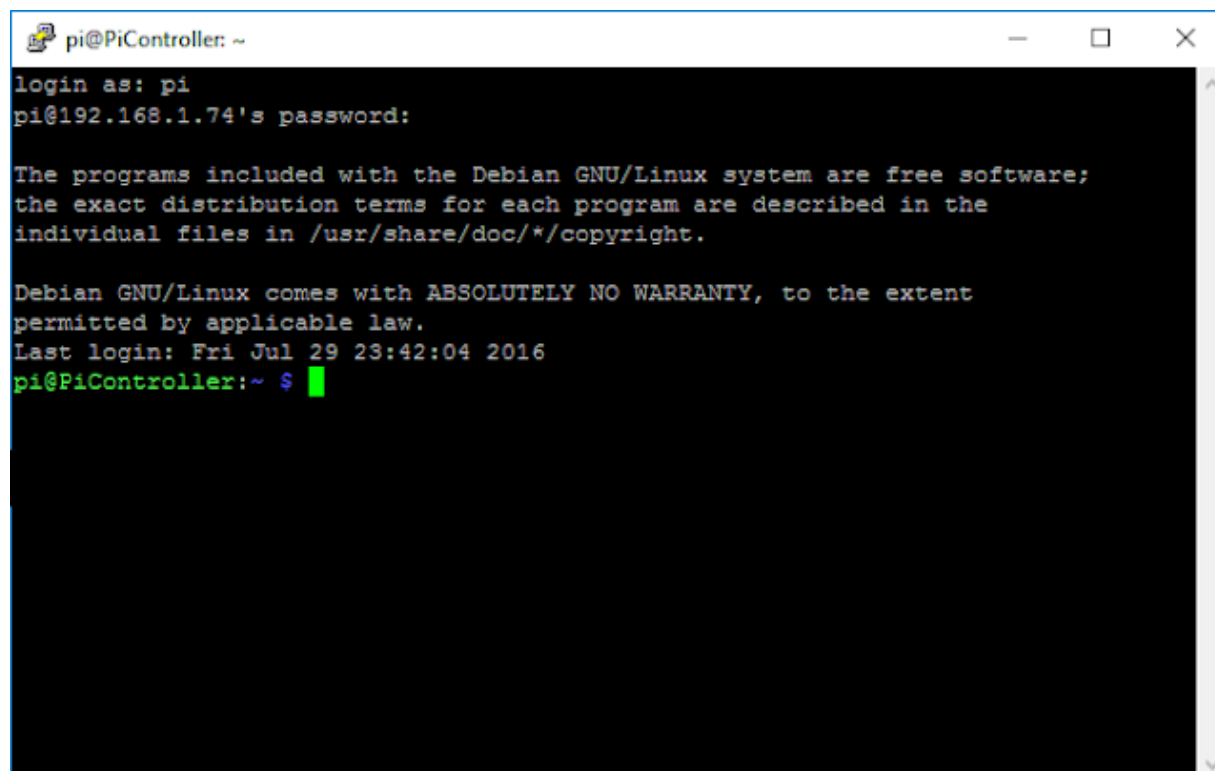
to spot in your network. Once you ping it, you will be able to see the ip address of the device. Save this IP address for later as we will use it in PuTTY.

Launch **PuTTY** and type the IP address of the RaspberryPi:



You should see something similar to the image below:





```
pi@PiController: ~
login as: pi
pi@192.168.1.74's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

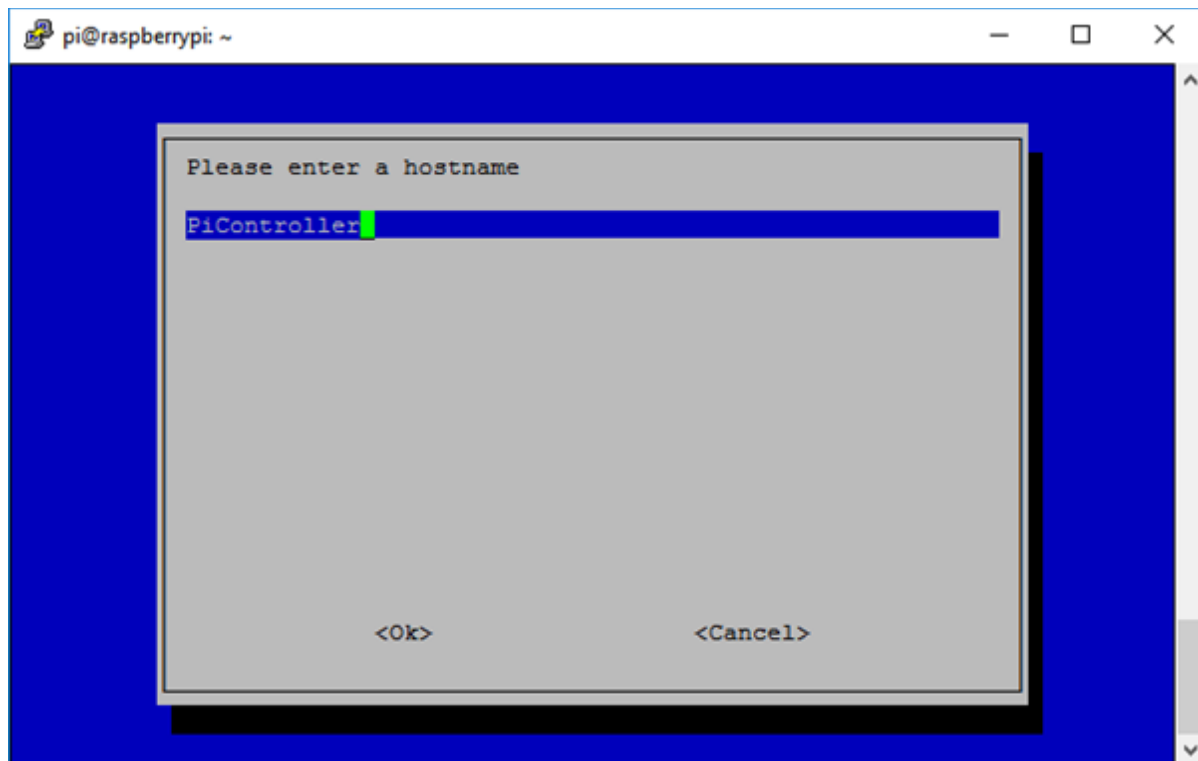
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 29 23:42:04 2016
pi@PiController:~ $
```

login as: **pi** and password: **raspberrypi** (each Rpi uses same login/password)

Type: **sudo raspi-config** to configure our device:

1. Go to Expand File System
2. Go to Advanced Options -> HostName -> set it to **PiController**
3. Go to Advanced Options -> MemorySplit -> set it to 16.
4. Go to Advanced Options -> SSH -> Enable.
5. Finish and leave the configuration.





Now we can start installing **MPICH3** and **MPI4PY**. Notice that these steps take a while (> 4h) so arrange some free time for this beforehand:

## Installing MPICH3

Follow the steps below to install version 3.2 of MPICH:

```
1  # *****
2  # These steps will allow you to install MPICH3 to your Raspbian Jessie distro
3  # Jordi Corbilla
4  # *****
5
6  # update the system
7  sudo apt-get update
8
9  # update packages
10 sudo apt-get dist-upgrade
11
12 # create the folder for mpich3
13 sudo mkdir mpich3
14 cd ~/mpich3
15
16 # download the version 3.2 of mpich
17 sudo wget http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz
```



```
18
19 # unzip it
20 sudo tar xzf mpich-3.2.tar.gz
21
22 # create folders for mpi
23 sudo mkdir /home/rpimpi/
24 sudo mkdir /home/rpimpi/mpi-install
25 mkdir /home/pi/mpi-build
26
27 # install gfortran
28 sudo apt-get install gfortran
29
30 # configure and install mpich
31 sudo /home/pi/mpich3/mpich-3.2/configure --prefix=/home/rpimpi/mpi-install
32 sudo make
33 sudo make install
34
35 # edit the bash script using nano editor that runs everytime the Pi starts
36 cd ..
37 nano .bashrc
38
39 # Add the following to the end of the file
40 # PATH=$PATH:/home/rpimpi/mpi-install/bin
41 # to save the details press "CTRL + ^x" -> using CTRL 6 and x
42 # press "y" and hit enter to leave.
43
44 # Reboot the Pi
45 sudo reboot
46
47 # Test that MPI works
48 mpiexec -n 1 hostname
```

**mpich3install.sh** hosted with ❤ by **GitHub**

[view raw](#)

Once you've got everything installed you should see something like the image below:



```
pi@PiController: ~
login as: pi
pi@192.168.1.74's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jul 30 18:40:10 2016
pi@PiController:~ $ mpiexec -n 1 hostname
PiController
pi@PiController:~ $
```

## Installing MPI4PY

Follow the steps below to install version 2.0 of MPI4PY:

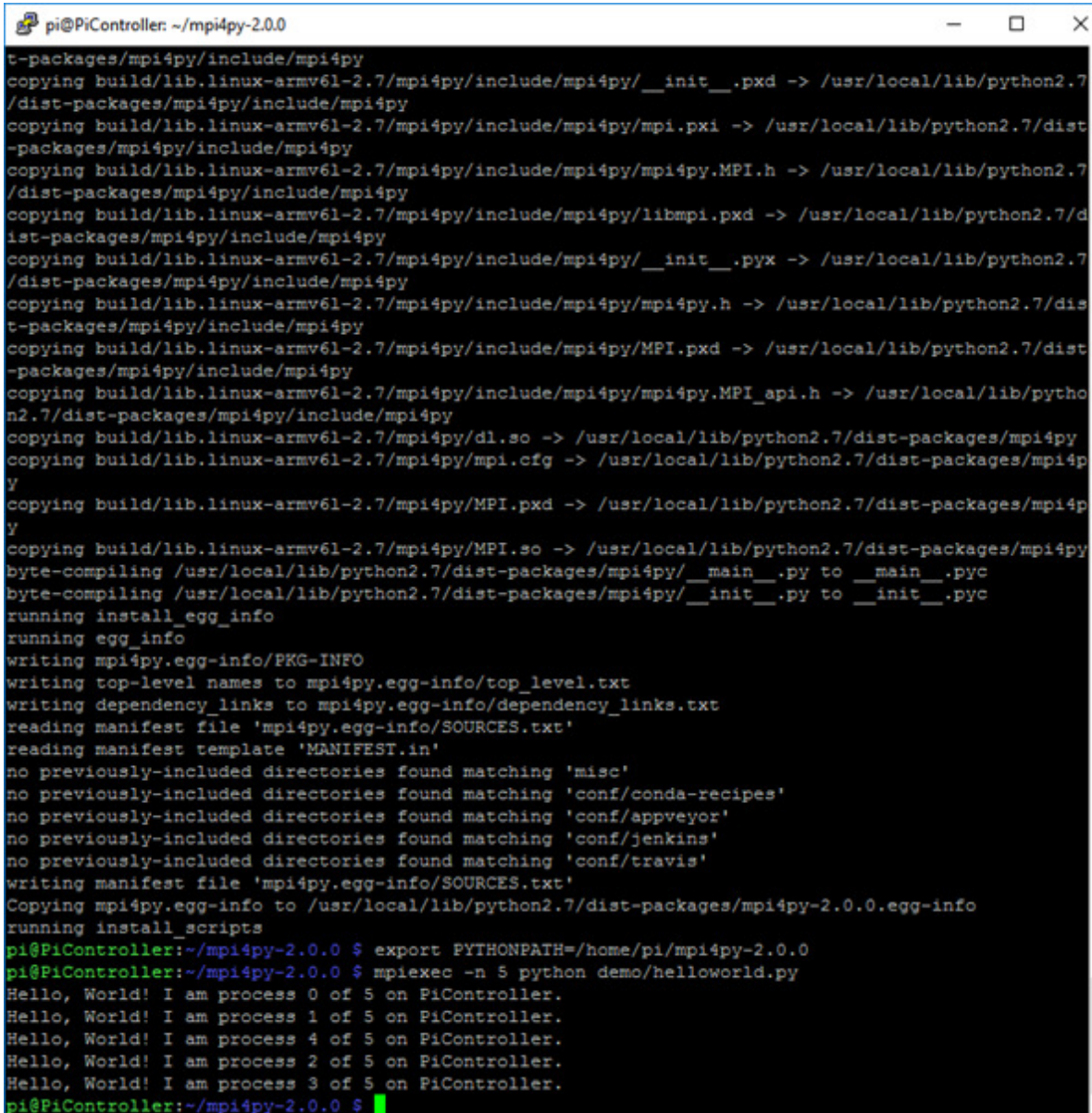
```
1  # *****
2  # These steps will allow you to install MPI4PY to your Raspbian Jessie distro
3  # Jordi Corbilla
4  # *****
5
6  # download mpi4py
7  wget https://bitbucket.org/mpi4py/mpi4py/downloads/mpi4py-2.0.0.tar.gz
8
9  #unzip the file
10 sudo tar -zxf mpi4py-2.0.0.tar.gz
11
12 # go to the directory
13 cd mpi4py-2.0.0
14
15 # install python-dev package
16 sudo aptitude install python-dev
17
18 # run the setup
19 python setup.py build
20 sudo python setup.py install
```

```
21
22 # Set the python path
23 export PYTHONPATH=/home/pi/mpi4py-2.0.0
24
25 # Test that MPI works on your device
26 mpiexec -n 5 python demo/helloworld.py
```

mpi4pyinstall.sh hosted with ❤ by GitHub

[view raw](#)

once installed you should see something like the image below:

A terminal window titled 'pi@PiController: ~/mpi4py-2.0.0' showing the installation of mpi4py. The output includes copying various files to /usr/local/lib/python2.7/dist-packages, byte-compiling \_\_main\_\_.py and \_\_init\_\_.py, and running the install script. After setting the PYTHONPATH, the command 'mpiexec -n 5 python demo/helloworld.py' is executed, resulting in five 'Hello, World!' messages from different processes on the PiController.

```
pi@PiController: ~/mpi4py-2.0.0
t-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/__init__.pxd -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/mpi.pxi -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/mpi4py.MPI.h -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/libmpi.pxd -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/__init__.pyx -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/mpi4py.h -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/MPI.pxd -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/include/mpi4py/mpi4py.MPI_api.h -> /usr/local/lib/python2.7/dist-packages/mpi4py/include/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/dl.so -> /usr/local/lib/python2.7/dist-packages/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/mpi.cfg -> /usr/local/lib/python2.7/dist-packages/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/MPI.pxd -> /usr/local/lib/python2.7/dist-packages/mpi4py
copying build/lib.linux-armv6l-2.7/mpi4py/MPI.so -> /usr/local/lib/python2.7/dist-packages/mpi4py
byte-compiling /usr/local/lib/python2.7/dist-packages/mpi4py/__main__.py to __main__.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/mpi4py/__init__.py to __init__.pyc
running install_egg_info
running egg_info
writing mpi4py.egg-info/PKG-INFO
writing top-level names to mpi4py.egg-info/top_level.txt
writing dependency links to mpi4py.egg-info/dependency_links.txt
reading manifest file 'mpi4py.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
no previously-included directories found matching 'misc'
no previously-included directories found matching 'conf/conda-recipes'
no previously-included directories found matching 'conf/appveyor'
no previously-included directories found matching 'conf/jenkins'
no previously-included directories found matching 'conf/travis'
writing manifest file 'mpi4py.egg-info/SOURCES.txt'
Copying mpi4py.egg-info to /usr/local/lib/python2.7/dist-packages/mpi4py-2.0.0.egg-info
running install_scripts
pi@PiController:~/mpi4py-2.0.0 $ export PYTHONPATH=/home/pi/mpi4py-2.0.0
pi@PiController:~/mpi4py-2.0.0 $ mpiexec -n 5 python demo/helloworld.py
Hello, World! I am process 0 of 5 on PiController.
Hello, World! I am process 1 of 5 on PiController.
Hello, World! I am process 4 of 5 on PiController.
Hello, World! I am process 2 of 5 on PiController.
Hello, World! I am process 3 of 5 on PiController.
pi@PiController:~/mpi4py-2.0.0 $
```

Now we have finished configuring the first RPi. Believe it or not if you reach this step and everything is working you should be proud of it. Now we will have to clone this SD card and place them into the other RPi's.

## Preparing the other RPi's

As mentioned in the step above, bring the SD card to your main computer and save the content of the SD card using Win32DiskImager. Now copy this new image to the other SD cards. **You should have now 4 SD cards with the same image.** As now we have 4 cloned SD cards, my advice is to plug every Rpi individually and change the host name of every new added Rpi into the network, e.g. pi01, pi02, pi03, etc.

Do the following for every new RPi added into the network:

**pi01:**

scan the network for a newly added device to find its IP address using a [network scanner](#). Once found, use PuTTY to access it and use the commands below to set it up:

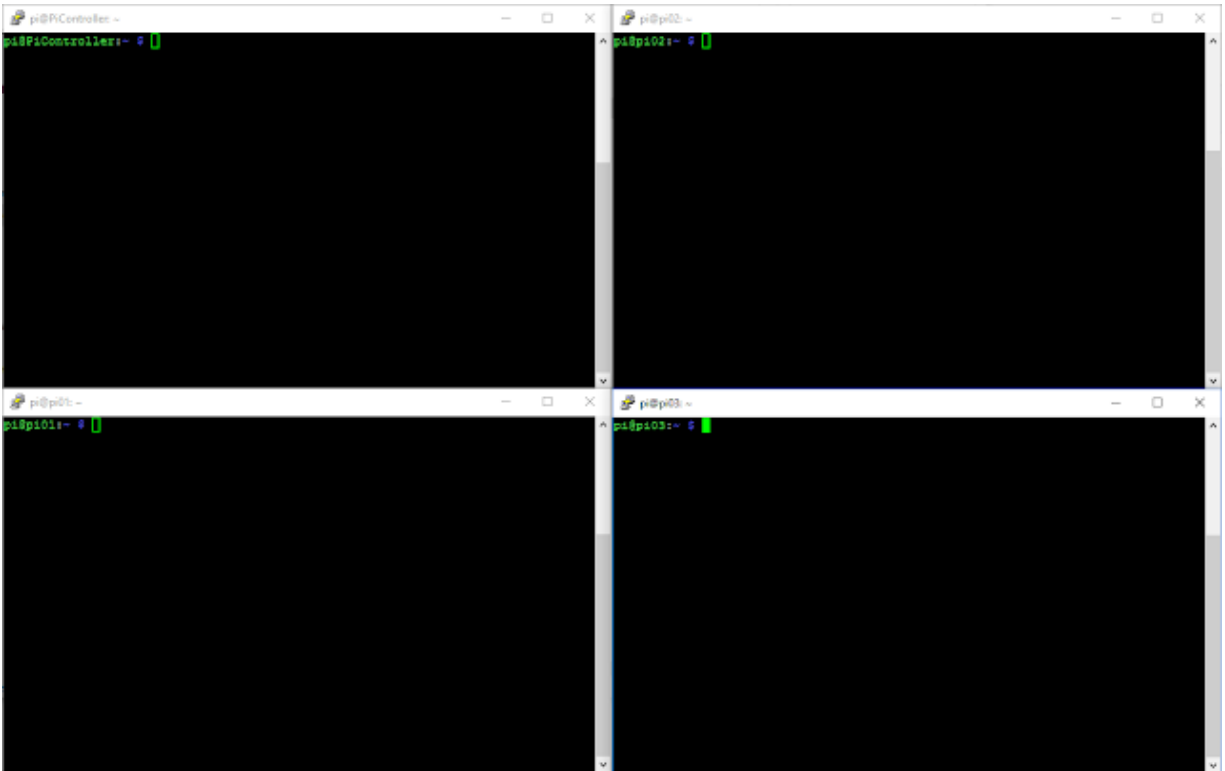
Type: **sudo raspi-config** to configure our device:

1. Go to Expand File System
2. Go to Advanced Options -> HostName -> set it to **pi01**
3. Go to Advanced Options -> MemorySplit -> set it to 16.
4. Go to Advanced Options -> SSH -> Enable.
5. Finish and leave the configuration.
6. sudo reboot.

Do the same for **pi02** and **pi03**. *Note that you can name your RPis the way you want.*

Once done you should be able to see them all 4 using PuTTY:



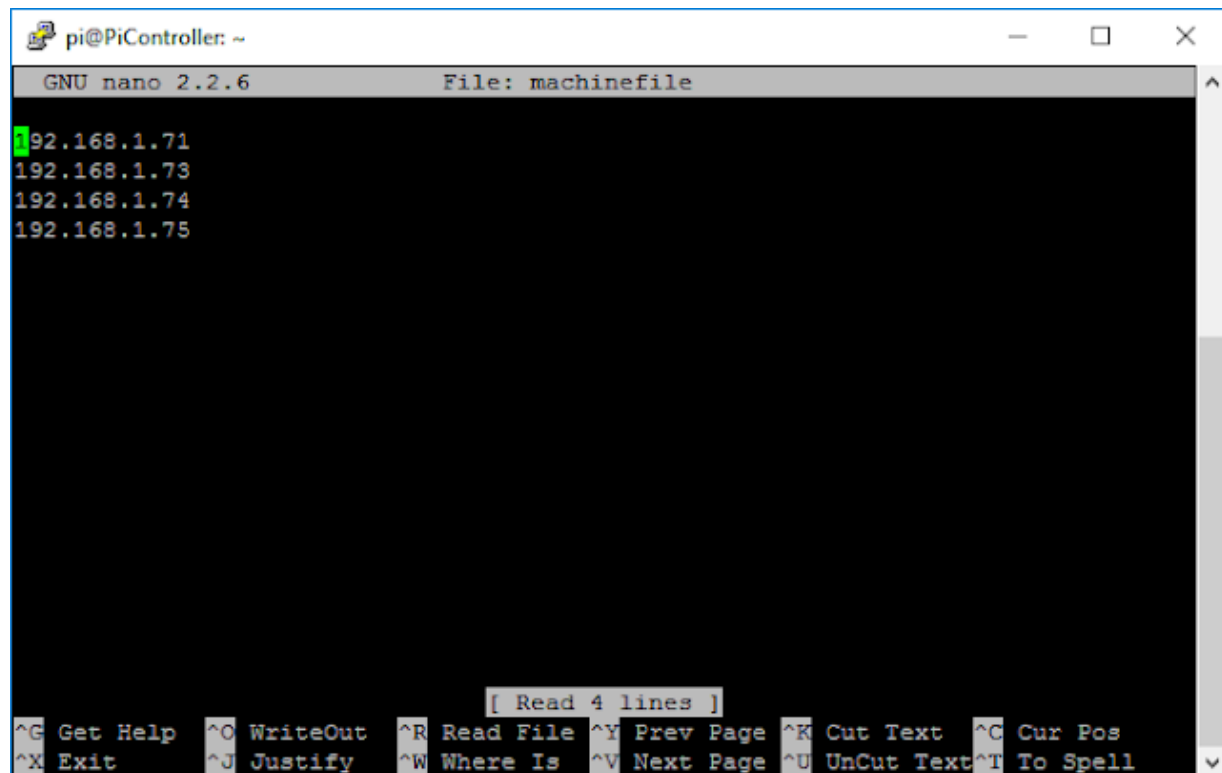


Once completed, each Rpi will have its own IP. We need now to store each IP address into a host file also known as machinefile. This file contains the hosts which to start the processes on.

Go to your first RPi and type:

**nano machinefile**

and add the following IP addresses: (Note that you will have to add your own):

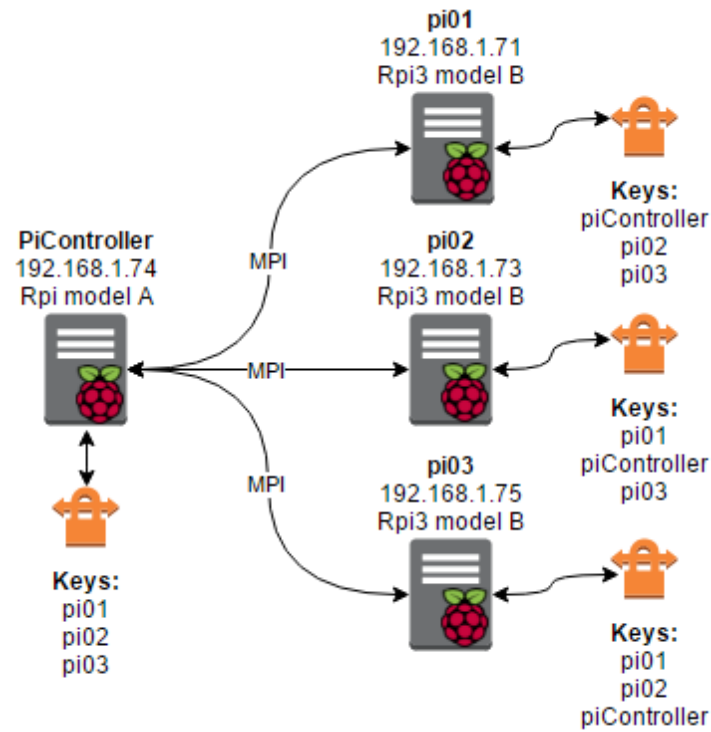


```
pi@PiController: ~  
GNU nano 2.2.6 File: machinefile  
192.168.1.71  
192.168.1.73  
192.168.1.74  
192.168.1.75  
[ Read 4 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

This will be used by the MPICH3 to communicate and send/receive messages between various nodes.

## Configuring SSH keys for each RPi

Now we need to be able to command each RPi without using users/passwords. To do this we will have to generate SSH keys for each RPi and then share each key to each device under authorised devices. This is the way MPI will be able to talk to each device without worrying about credentials. This process is a bit tedious but once completed you will be able to run MPI without problems.



Run the following commands from the first Pi (PiController):

```

1  # *****
2  # These steps will allow you to configure SSH on every device
3  # Run the commands below from PiController device
4  # Jordi Corbilla
5  # *****
6
7  # PiController (192.168.1.74)
8  ssh-keygen
9  cd ~
10 cd .ssh
11 cp id_rsa.pub PiController
12
13 # pi01 (192.168.1.71)
14
15 ssh pi@192.168.1.71
16 ssh-keygen
17 cd .ssh
18 cp id_rsa.pub pi01
19 scp 192.168.1.74:/home/pi/.ssh/PiController .
20 cat PiController >> authorized_keys
21 exit
22
23 # pi02 192.168.1.73
24
    
```



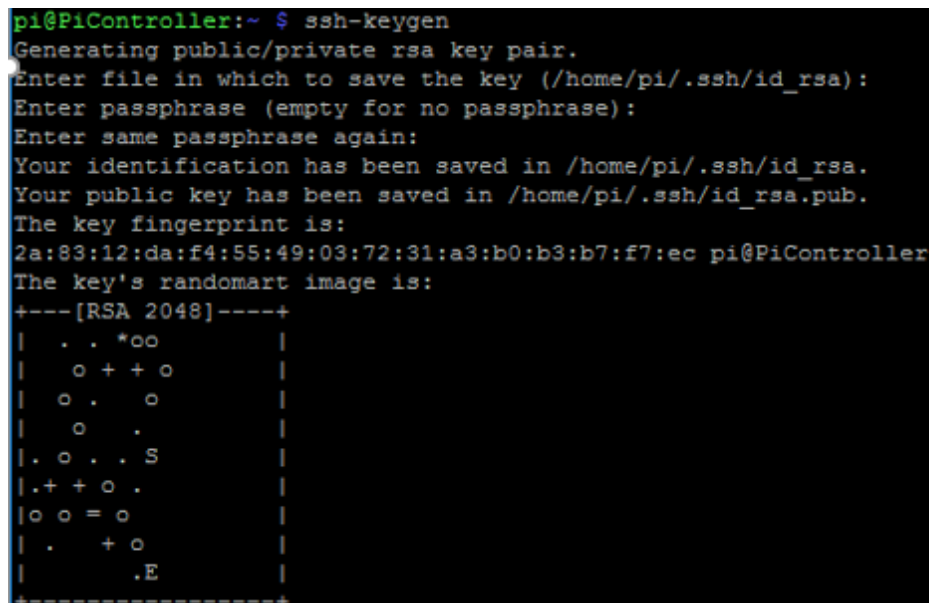
```

25  ssh pi@192.168.1.73
26  ssh-keygen
27  cd .ssh
28  cp id_rsa.pub pi02
29  scp 192.168.1.74:/home/pi/.ssh/PiController .
30  cat PiController >> authorized_keys
31  exit
32
33  # pi03 192.168.1.75
34
35  ssh pi@192.168.1.75
36  ssh-keygen
37  cd .ssh
38  cp id_rsa.pub pi03
39  scp 192.168.1.74:/home/pi/.ssh/PiController .
40  cat PiController >> authorized_keys
41  exit

```

sshKeysRpi.sh hosted with ❤ by [GitHub](#)

[view raw](#)



```

pi@PiController:~ $ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pi/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pi/.ssh/id_rsa.
Your public key has been saved in /home/pi/.ssh/id_rsa.pub.
The key fingerprint is:
2a:83:12:da:f4:55:49:03:72:31:a3:b0:b3:b7:f7:ec pi@PiController
The key's randomart image is:
+---[RSA 2048]---+
|  . . *oo      |
|  o + + o      |
|  o . o        |
|  o .          |
|. o . . S      |
|. + + o .      |
|o o = o        |
|  . + o        |
|      .E       |
+-----+

```

When running the ssh-keygen just hit enter (if you don't want to add specific passphrase) and the RSA key will be generated for you automatically.

Now we have configured the link between **PiController** to every single device but we still need to configure the other way around. So you will have to run the following commands from every individual device:

```

1  # *****
2  # These steps will allow you to configure SSH on every device

```

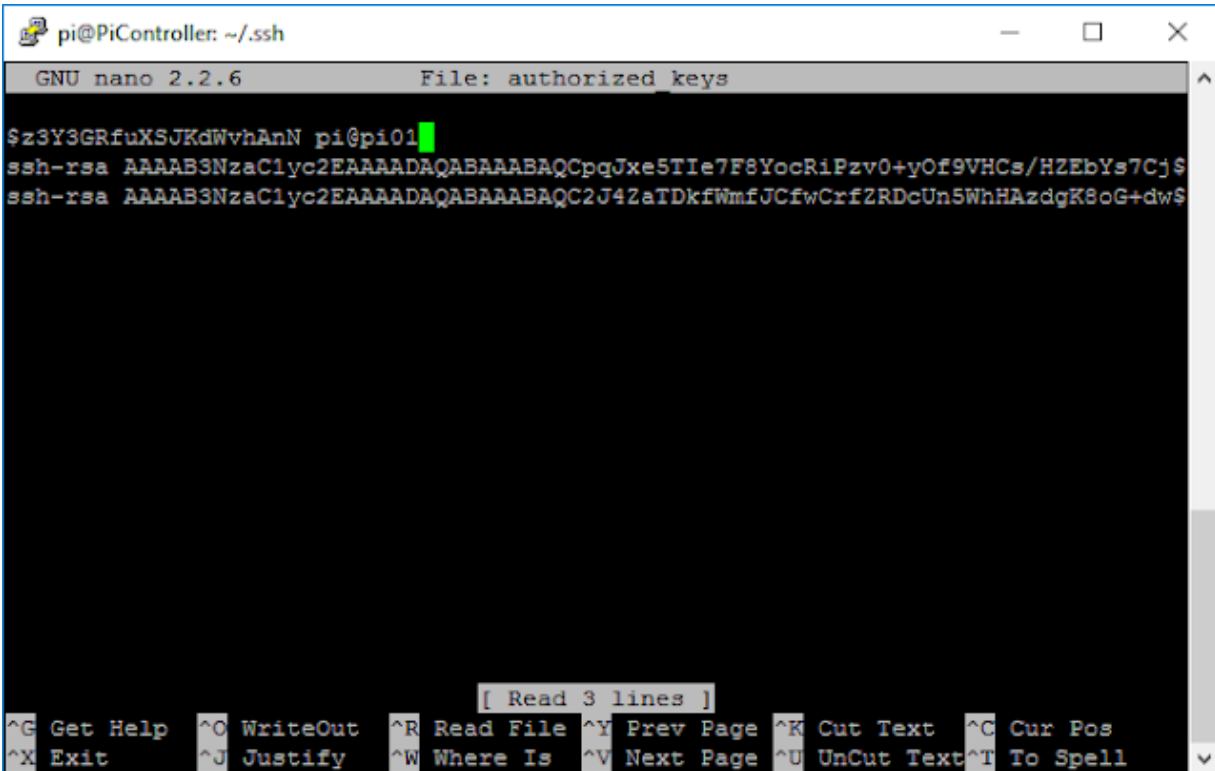
```
3 # Jordi Corbilla
4 # *****
5
6 # run this from PiController using PuTTY
7 cd ~
8 cd .ssh
9 scp 192.168.1.71:/home/pi/.ssh/pi01 .
10 cat pi01 >> authorized_keys
11 scp 192.168.1.73:/home/pi/.ssh/pi02 .
12 cat pi02 >> authorized_keys
13 scp 192.168.1.75:/home/pi/.ssh/pi03 .
14 cat pi03 >> authorized_keys
15
16 # run this from pi01 using PuTTY
17 cd ~
18 cd .ssh
19 scp 192.168.1.73:/home/pi/.ssh/pi02 .
20 cat pi02 >> authorized_keys
21 scp 192.168.1.75:/home/pi/.ssh/pi03 .
22 cat pi03 >> authorized_keys
23
24 # run this from pi02 using PuTTY
25 cd ~
26 cd .ssh
27 scp 192.168.1.71:/home/pi/.ssh/pi01 .
28 cat pi01 >> authorized_keys
29 scp 192.168.1.75:/home/pi/.ssh/pi03 .
30 cat pi03 >> authorized_keys
31
32 # run this from pi03 using PuTTY
33 cd ~
34 cd .ssh
35 scp 192.168.1.71:/home/pi/.ssh/pi01 .
36 cat pi01 >> authorized_keys
37 scp 192.168.1.73:/home/pi/.ssh/pi02 .
38 cat pi02 >> authorized_keys
39
40 # Inspect each authorized_keys file on each device and you will see the keys there for every device
```

sshKeysIndividualRPi.sh hosted with ❤ by **GitHub**

[view raw](#)

open the authorized\_keys files and you will see the additional keys there. Each

authorized\_keys file on each device should contain 3 keys (as stated in the architecture diagram above).



```
pi@PiController: ~/.ssh
GNU nano 2.2.6 File: authorized_keys

$z3Y3GRfuXSJKdWvhAnN pi@pi01
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCPqJxe5TTe7F8YocRiPzv0+yOf9VHCs/HZEbYs7Cj$
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCC2J4ZaTDkfWmfJCfwCrFzRDcUn5WhHAzdgK8oG+dw$

[ Read 3 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Now the system is ready for testing.

Note that if your IP address changes, the **keys** will not be valid and the steps will have to be repeated.

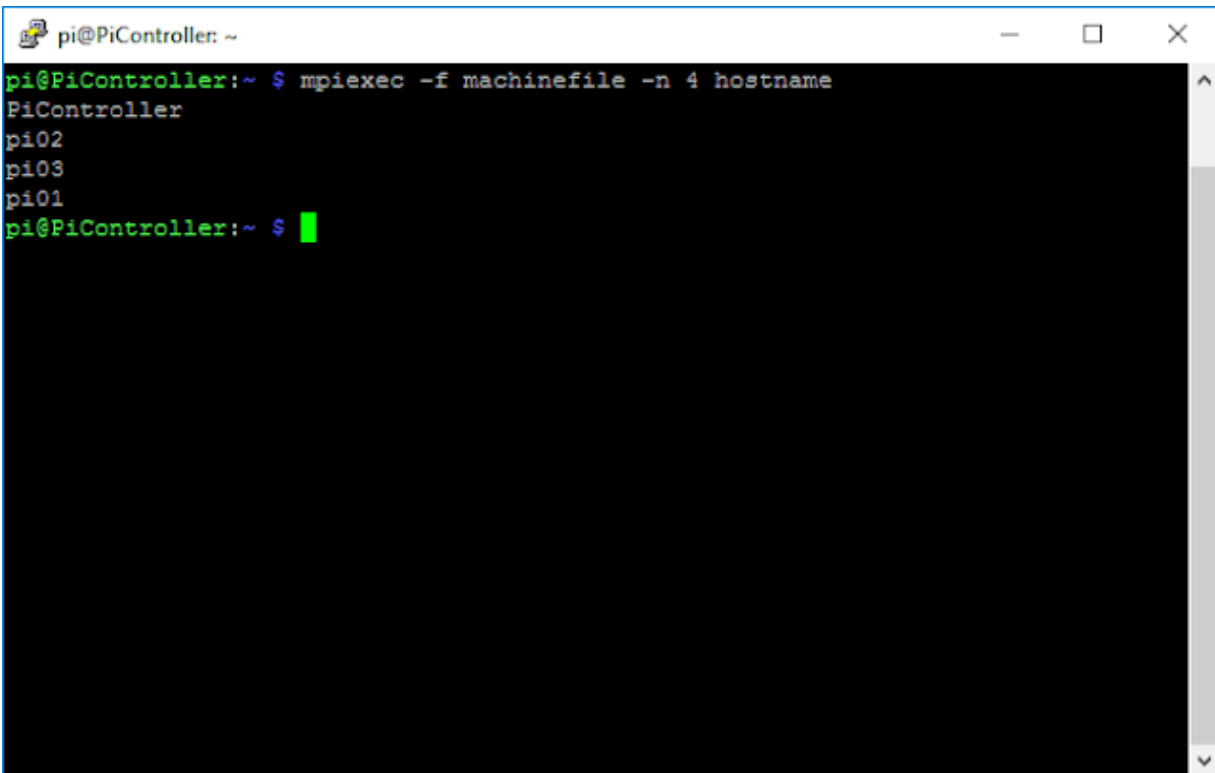
### Testing the cluster

At this point I will just include a small example for you to test that the cluster works as expected. Later on I will publish a more complex scenario with a refined configuration to maximise the power of the cluster.

If everything is configured correctly, the following command should work correctly:

**mpirun -f machinefile -n 4 hostname**



A terminal window titled 'pi@PiController: ~' with standard window controls. The prompt is 'pi@PiController:~ \$'. The command 'mpiexec -f machinefile -n 4 hostname' has been entered. The output shows 'PiController' followed by three lines of 'pi02', 'pi03', and 'pi01' on separate lines. The prompt 'pi@PiController:~ \$' is visible again with a green cursor.

```
pi@PiController:~ $ mpiexec -f machinefile -n 4 hostname
PiController
pi02
pi03
pi01
pi@PiController:~ $
```

You can see that each Device has replied back and every key is used without problems.

Now run the following command to test a helloworld example:

```
mpiexec -f machinefile -n 4 python /home/pi/mpi4py-2.0.0/demo
/helloworld.py
```

You should see something like the image below:



```

pi@PiController: ~
pi@PiController:~ $ mpiexec -f machinefile -n 4 python /home/pi/mpi4py-2.0.0/dem
o/helloworld.py
Hello, World! I am process 0 of 4 on pi01.
Hello, World! I am process 1 of 4 on pi02.
Hello, World! I am process 3 of 4 on pi03.
Hello, World! I am process 2 of 4 on PiController.
pi@PiController:~ $

```

Now our system is **ready** to take any parallel computing application we want to develop.

Watch this space for more!.

### Next steps

I will be creating more complex scenarios and squeezing the architecture to test its limits. Soon more!. Give it a go and let me know if you face any problem during the set up.

Jordi

Share This:  Facebook  Twitter  Google+

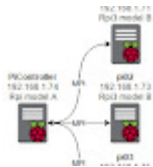


Jordi Corbilla

I am a full stack [Software Architect](#) and I consider myself a problem solver with the ability of getting things to work. Having a keen eye on quality,

architecture and risks this lets me build good software. I am mainly interested in Delphi, .NET, Databases, AI, compilers, grammars, graphics and more mathematical stuff. If you like this page you could also visit me on twitter [@thunderjordi](#) and on [Facebook](#).

## Related Posts:



[Creating a Raspberry Pi 3 Cluster - "Supercomputer", for parallel computing.](#)

In this quick article I will show you how to create your own Raspberry Pi cluster for parallel computing via MPI (Messaging Passing Interface) li... [Read More](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

60 comments:

**Herbert Sauro** 4 August 2016 at 02:44

I'll be interested to see how you get on with this, especially what kind of performance you get.

[Reply](#)

[Replies](#)



**Jordi Corbilla** 4 August 2016 at 23:29

Hi Herbert,

Sure, I'm still in the coding phase testing the cluster. Soon I'll share my results.

Cheers,  
Jordi



## Reply



**akshay pai** 7 April 2017 at 19:33

Great article. I had built one using 70 Raspberry Pis as my final year college project but they were made from Raspberry pi B+ models and we faced a lot of issues: [sourcedexter.com/6-common-errors-when-building-a-raspberry-pi-supercomputer/](http://sourcedexter.com/6-common-errors-when-building-a-raspberry-pi-supercomputer/) . However, we succeeded in building it and also ran a multi-document text summarization algorithm on it.

[Reply](#)

**Anonymous** 25 May 2017 at 20:29

Thank you for this tutorial.

I'm having a problem with the line

```
scp 192.168.1.74:/home/pi/.ssh/PiController
```

when I try to use that commmand I end up with

```
usage: scp [-12346BCpqrv] [-c cipher] etc. etc.
```

Any advice?

Thanks!

[Reply](#)

[Replies](#)





**Anonymous** 1 June 2017 at 04:27

I forgot the . at the end, it works when you include it.

```
scp 192.168.1.74:/home/pi/.ssh/PiController .
```

[Reply](#)

**Anonymous** 4 June 2017 at 18:42

Good tutorial, hi i got a problem when i write the command ping raspberrypi, and it doesn't recognized this, any advice

[Reply](#)

[Replies](#)



**Jordi Corbilla** 7 June 2017 at 11:18

you mean that it doesn't recognise the ping command? you might have to make sure your system32 folder is part of your path.

[Reply](#)

**Ethernet Cables** 7 July 2017 at 10:52

The article you have shared here very awesome. I really like and appreciated your work. I read deeply your article, the points you have mentioned in this article are useful

[Reply](#)



## Replies



**Jordi Corbilla** 7 July 2017 at 12:35

Many thanks. much appreciated.

## Reply



**Michael Goodwin** 23 July 2017 at 06:36

Hi Jordi, very interested in this and keen to get started. The goal is less about parallel computing and more about having GNS3 on a cluster for my CCNA studies. I found this <https://www.gns3.com/discussions/the-worlds-first-gns3-beowulf-cl> and have had all kinds of dramas getting MPICH and MPI4PY installed. So I'll give this a go. If I get GNS 3 installed as a step just prior to the image creation/duplication.. would you expect any/many issues? I've posted questions on Jason's forum but alas he seemed to have dropped off the face of the earth in 2015 lol

## Reply

## Replies



**Jordi Corbilla** 25 July 2017 at 16:57

Hi Michael,

I don't expect many issues but you'll have to check the version and its dependencies as GNS 3 might require a different version that I used during my testing.

Cheers,



Jordi

## Reply



**Michael Goodwin** 23 July 2017 at 12:08

Hi, I get to the step where I install MPI4PI and this throws a spanner in the works: python setup.py build. It gives:

Assembler messages:

Fatal error: can't create build/temp.linux-armv7l-2.7/src/MPI.o:  
Permission denied

error: command '/home/rpimpi/mpi-install/bin/mpicc' failed with exit status 1

[Reply](#)

## Replies



**Jordi Corbilla** 25 July 2017 at 17:01

weird, are you running as root? Just make sure that you have permissions to write. It sounds like you don't have enough permissions there.

cheers,

Jordi



**Michael Goodwin** 30 July 2017 at 11:07

Hi dude, yeah I just ch pwned that the mpi4py-2.0.0 and it all went swimmingly.

Next step is to get GNS3 on one, then continue the rest.. then copy the image to the others and



see how we go



**Jordi Corbilla** 31 July 2017 at 10:44

Excellent great to hear!

[Reply](#)

**Anonymous** 24 July 2017 at 03:53

Hello Jordi at what point should I plug my other pis into my network switch ?

[Reply](#)

[Replies](#)



**Jordi Corbilla** 25 July 2017 at 17:00

Once you have cloned the SD cards, you can just keep plugging them in one by one and change the IP address on each one prior to connecting the next one. That should do.

Cheers,  
Jordi

[Reply](#)



**Michael Goodwin** 31 July 2017 at 10:46

I know this is kind of unrelated to the mpich thing [got that k working].. Anyone know how/where to get qemu-kvm for armhf [^](#)

from? I've tried many places and `sudo apt-get instal qemu-kvm`  
[prereq for gns3-server] i get:

Package `qemu-kvm` is not available, but is referred to by another package.

This may mean that the package is missing, has been obsoleted,  
or  
is only available from another source

E: Package '`qemu-kvm`' has no installation candidate

Any ideas?

[Reply](#)

#### [Replies](#)



**Jordi Corbilla** 31 July 2017 at 12:18

I don't know if this package is available for ARM. Have you tried `kvmtool`? It's an alternative: <https://github.com/clearlinux/kvmtool>

[Reply](#)



**Christinat Damien** 27 August 2017 at 00:20

hi.. i got a small problem. all works fine exept this : when i plug a usb disk or other on node 2,3 .. master node dosen't will see it. is there a fix for it ?

sorry for my english but i live in switzerland and i never has learn it...

thanks for response

[Reply](#)



## Replies



**Jordi Corbilla** 2 October 2017 at 10:52

Hi Cristinat,

this might be a problem with that USB, you might have to enable the USB form the Pi.

Cheers,  
Jordi

## Reply



**Unknown** 1 October 2017 at 10:30

can i use this to run SETI@home (BOINC).  
Does the BOINC Client use all 4 Pis?

Reply



**Peter Wildbichler** 1 October 2017 at 10:31

can i use this to run SETI@home (BOINC).  
Does the BOINC Client use all 4 Pis?

Reply

## Replies



**Jordi Corbilla** 2 October 2017 at 10:51

Hi Peter,

I don't know if BOINC is able to get the cluster



altogether. You might have to run 4 individual BOINC's and then find a way to make the hardware available in some way.

Cheers,  
Jordi

## Reply

**Anonymous** 3 October 2017 at 06:24

Hello I've been going over this tutorial with 3 of my Raspberry Pis. I'm having an issue with `authorized_keys`. I have created them for each pi and each pi knows the key for the others. I've confirmed the keys are identical just in case.

However, when checking if the configuration was successful with `mpi`, `PiController` works, but `Pi01` and `Pi02` are access denied.

Was wondering if something was missing or perhaps I did something wrong?

## Reply

## Replies

**Anonymous** 3 October 2017 at 11:38

Never mind. I realized I used the `PiController`, `Pi01` and `Pi02` files, not the `.pub` files



**Jordi Corbilla** 3 October 2017 at 12:18

Cool that you could solve it. Usually it's





something like that as it's quite straightforward.

Cheers,  
Jordi

### Reply



**Unknown** 3 October 2017 at 12:31

For `mpirun -f machinefile -n 4 hostname` I get...

PiController  
Pi01

But Pi02 is never reached and eventually times out. However I can  
ssh to it through PiController and Pi02 can ssh to PiController.

Any help in regards to this?

### Reply

### Replies



**Jordi Corbilla** 4 October 2017 at 13:42

can you ping Pi02?

### Reply



**Unknown** 8 October 2017 at 22:33

Hey, This is a great tutorial, absolutely love it, thanks man



[Reply](#)

[Replies](#)



**Jordi Corbilla** 9 October 2017 at 10:38

Thanks, really happy you found it useful!

Cheers,  
Jordi

[Reply](#)



**cyril Marteau** 16 November 2017 at 00:04

Hello Jordi,

First of all, congrats for this post !

I have a project with a friend. Building a cluster of computer with an x86 architecture and we w'de like the Rpi as frontend. Do you think this might work ? If I install debian with mpich3 and openmpi packages on the computers ?

Thank you in advance for your answer.

Best regards

[Reply](#)



**cyril Marteau** 16 November 2017 at 00:06

Hello Jordi,

Congrats for your post !

We have a project with a friend. To build a cluster of computers but we would like the raspberry pi 3 as frontend. Do you think  
might be possible ?



If we install the two packages mpich3 and openmpi ?

Thank you in advance for your answer !

Cyril

[Reply](#)

### Replies



**Jordi Corbilla** 17 November 2017 at 13:12

Hi Cyril,  
Yes, it should work.

Cheers,  
Jordi

[Reply](#)

**Anonymous** 16 November 2017 at 07:50

Hi, great article and would love to put something like this together. Wanted to point out that the ssh keys aren't tied to IPs so you can reuse the keys even if IP changes. If it didn't work for you, it may have been due to the Host ssh keys changing. This happens when you reinstall the OS, but the individual user keys shouldn't be affected. Just make sure to clear out `~pi/.ssh/known_hosts` file and you'll be good to go.

[Reply](#)



**Helmut Hauck** 16 November 2017 at 19:41

Hi

Really an excellent tutorial. Thank you very much.

[Reply](#)

[Replies](#)



**Jordi Corbilla** 17 November 2017 at 13:11

Many thanks Helmut! much appreciated!

[Reply](#)



**Salvatore Mandaglio** 26 November 2017 at 12:47

Hi

Thank you for this tutorial. I followed every step and everything worked fine until the final cluster test. When i try to execute `mpiexec -f machinefile -n 4 hostname` i'm getting following error. Do you have any idea what i did wrong?

```
[mpiexec@PiController] HYDU_parse_hostfile (/home/pi/mpich3
/mpich-3.2/src/pm/hydra/utils/args/args.c:319): unable to open
host file: machinefile
```

```
[mpiexec@PiController] mfile_fn (/home/pi/mpich3/mpich-3.2/src
/pm/hydra/ui/mpich/utils.c:336): error parsing hostfile
```

```
[mpiexec@PiController] match_arg (/home/pi/mpich3/mpich-
3.2/src/pm/hydra/utils/args/args.c:152): match handler returned
error
```

```
[mpiexec@PiController] HYDU_parse_array (/home/pi/mpich3
/mpich-3.2/src/pm/hydra/utils/args/args.c:174): argument matching
returned error
```

```
[mpiexec@PiController] parse_args (/home/pi/mpich3/mpic
3.2/src/pm/hydra/ui/mpich/utils.c:1596): error parsing input array
```

```
[mpiexec@PiController] HYD_uui_mpx_get_parameters (/home
/pi/mpich3/mpich-3.2/src/pm/hydra/ui/mpich/utils.c:1648):  unable
to parse user arguments
[mpiexec@PiController] main (/home/pi/mpich3/mpich-3.2/src
/pm/hydra/ui/mpich/mpiexec.c:153): error parsing parameters
```

Thanks in advance  
Salvi

[Reply](#)

### Replies



**Jordi Corbilla** 27 November 2017 at 14:36

Hi Salvi,  
Are you running this with elevated permissions?

Cheers,  
Jordi

[Reply](#)



**Armando Jipsion** 26 November 2017 at 18:09

Hello Jordi  
Congratulations a great article, install the cluster without problems.  
I would like to know if the c and c ++ libraries can be integrated  
into the cluster  
You could indicate me where to download other examples to run  
them in the cluster  
Thank you

[Reply](#)



## Replies



**Jordi Corbilla** 27 November 2017 at 14:36

Hi Armando, I don't know which libraries are you referring to.

Cheers,  
Jordi

## Reply

**Anonymous** 15 December 2017 at 11:30

Dear all, thank you for the very well done instructions. Just want to ask you how to start all the nodes, wich is the command. When i launch my command from the master "./cpuminer ..." the other nodes doesn't listen this command, it seems that only my master is working. Thank you for any help.

## Reply

## Replies

**Anonymous** 26 December 2017 at 00:48

Hi, correct machinefile: set PiController IP on the last line.

And then:

```
mpiexec -machinefile /home/pi/machinefile -n 3  
~/cpuminer-multi/./cpuminer ...
```

## Reply





**Jared McCart** 23 December 2017 at 12:24

Hello Mr. Corbilla, great tutoial. Only 1 of 3 tutorials that has allowed me to run MPI.

!!!PROBLEM!!!

INPUT:

```
mpixexec -f machinefile -n 4 hostname
```

OUTPUT:

```
mpixexec: Error: unknown option "-f"
```

```
Type 'mpixexec --help' for usage.
```

Please advise.

P.S. Happy Holidays in advance.

[Reply](#)

### Replies

**Anonymous** 26 December 2017 at 00:42

```
mpixexec -machinefile /home/pi/machinefile -n 4  
hostname
```



**Jordi Corbilla** 5 January 2018 at 11:56

thanks!



**Jared McCart** 8 January 2018 at 02:43

Thank you, but this only gives me one host name witch is my PiController.

INPUT

```
mpixexec -machinefile /home/pi/machinefile -n 4  
hostname
```





## OUTPUT

PiController

PiController

Picontroller

Picontroller

I have changed the order of IP addresses in the machinefile and it will only read the first IP and give that hostname 4 times.



**Jordi Corbilla** 8 January 2018 at 11:18

Hi Jared,

That behaviour is a bit weird. Can you test your machinefile with just 2 IPs? and see if the behaviour is the same?

Cheers,

Jordi

### Reply



**DIWAKAR DIWEDI** 4 January 2018 at 11:07

This comment has been removed by the author.

### Reply

### Replies



**Jordi Corbilla** 5 January 2018 at 11:53

Hi Diwakar,



Have you configured correctly MPI on those nodes? You should only get 1 output in the master node.

Cheers,  
Jordi



**DIWAKAR DIWEDI** 10 January 2018 at 14:04

This comment has been removed by the author.

[Reply](#)

**Anonymous** 24 January 2018 at 14:37

Hi,

What is the difference between a Beowulf cluster and an MPI cluster?

[Reply](#)



**Shubham** 19 February 2018 at 12:32

Hi

First of all a great tutorial. I simply copy pasted it for further use. I am planning to build a 32 node nano super computer using raspberry pi3B models. My aim is not just to try out the rig for computational fluid dynamics but also for running certain codes through my web browser, say firefox. Right now I am in the process of buying the materials as a 32 node system will cost n about 100,000 INR. My main concern is as follows. 1. Can I use

Ubuntu OS for the master as well as the slave nodes? 2. If I open a web browser through the master node and run my program/codes inside the browser will it be able to use the combined CPU power of all the nodes?

Thanks in advance

[Reply](#)

### Replies



**Jordi Corbilla** 26 February 2018 at 16:10

Hi Shubham,

For what you are after you have to follow a different approach, with the current design it won't work. I'm working towards this idea so, when I have it done I will publish it.

Cheers,  
Jordi

[Reply](#)

**Anonymous** 7 March 2018 at 09:43

Hi Jordi,

great tutorial, got the cluster running. Still being a python and raspberry NOOB I was wondering if you could help me out. I got cpuminer-multi running on the main node only I want to run it on all nodes in my cluster, could you tell me how I could accomplish this?



Thanks,  
Simon

[Reply](#)

### Replies



**Jordi Corbilla** 7 March 2018 at 13:39

Hi Simon,

You'll have to look into it. That package has different settings so probably there is help online talking about it.

Cheers,  
Jordi

[Reply](#)

**Anonymous** 19 March 2018 at 18:57

Hi Jordi,

You should look at this article. Someone is using your work.  
<https://www.techworm.net/2018/03/learn-build-supercomputer-raspberry-pi-3-cluster.html>

[Reply](#)



**Ahmad Ridwan** 24 April 2018 at 06:40

Hi Jordi,



Great tutorial! I got my cluster running.

but can you give some of code or project that test performance of cluster (execution time) ?

sorry for my english.

Regards,

Ahmad Ridwan

[Reply](#)

**usb cables** 9 May 2018 at 07:35

Nice post, Thanks for sharing with us

[Reply](#)

**Anonymous** 19 September 2018 at 03:39

Yay, I got mine to work. I had slightly older software as I was following a set of instructions very similar to this so when it came to the final test, I had to change the mpi4py version to the one I am using and it worked. I am so happy.

[Reply](#)

Enter your comment...



Comment as: Doug's Theology E

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)



## About the Author



**Jordi Corbilla**



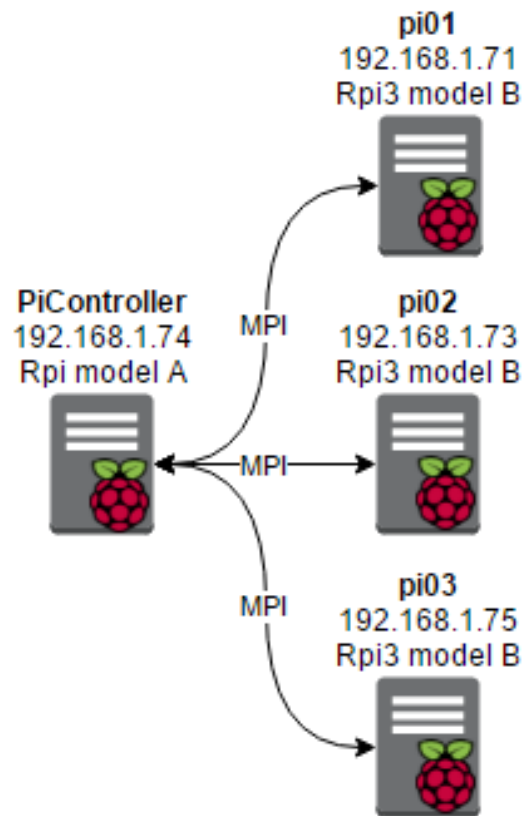
Follow

0

I am a full stack [Software Architect](#) and I consider myself a problem solver with the ability of getting things to work. Having a keen eye on quality, architecture and risks this lets me build good software. I am mainly interested in Delphi, .NET, Databases, AI, compilers, grammars, graphics and more mathematical stuff. If you like this page you could also visit me on twitter [@thunderjordi](#) and on [Facebook](#).

[View my complete profile](#)

(Messaging Passing ...



## Total Pageviews



1 4 3 5 2 8 1

## Featured post

[Creating a Raspberry Pi 3 Cluster - "Supercomputer", for parallel computing.](#)

In this quick article I will show you how to create your own Raspberry Pi cluster for parallel computing via MPI

## Recent Posts

0000562298



## Blog Archive

- ▶ [2018](#) (3)
- ▶ [2017](#) (7)
- ▼ [2016](#) (15)
  - ▶ [December](#) (2)
  - ▶ [November](#) (1)
  - ▶ [October](#) (1)
  - ▶ [September](#) (1)
  - ▶ [August](#) (2)
  - ▼ [July](#) (1)
    - [Creating a Raspberry Pi 3 Cluster - "Supercomputer..."](#)
- ▶ [June](#) (2)
- ▶ [May](#) (4)
- ▶ [March](#) (1)
- ▶ [2015](#) (5)
- ▶ [2014](#) (2)

- ▶ [2013](#) (4)
- ▶ [2012](#) (16)
- ▶ [2011](#) (42)
- ▶ [2010](#) (82)
- ▶ [2009](#) (285)
- ▶ [2008](#) (46)

## Google+ Followers

**Jordi Corbilla**

Add to circles

Followers list is private

## Subscribe!

SUBMIT

Donate



Copyright © 2019 [Random thoughts on coding and technology](#) | Powered by [Blogger](#)  
| All code samples included in this blog are licensed under a [Creative Commons Attribution 4.0 International License](#).  
Design by [CrestaProject](#) | Blogger Theme by [Lasantha - PremiumBloggerTemplates.com](#) |  
Distributed By [Gooyaabi Templates](#)

