

Bitacora:Cluster con MPI

Carlos Andres Acosta Ramos

Humberto de Jesus Flores Acuña

José Roberto Guerrero Zurita

25 Junio 2016

## 1. Requisitos

Estos son los programas necesarios para montar el cluster.

### 1.1. Servidor

- gcc (gfortran)
- ssh
- rpcbind
- nfs-kernel-server
- mpich3 en tar.gz<sup>1</sup>

### 1.2. Servidor

- gcc (gfortran)
- ssh
- rpcbind
- nfs-common
- mpich3 en tar.gz

---

<sup>1</sup>Se puede obtener de <https://www.mpich.org/downloads/>

## 2. Compilando mpich3

Una vez que hemos descargado **mpich-3.2.tar.gz** el siguiente paso es descomprimir el archivo en */home/pi*

```
1 $ tar -xzf mpich-3.2.tar.gz
2 $ cd mpich-3.2/
3 $ ./configure --prefix=/home/pi//mpich-install 2>&l | tee c.txt
4 $ make 2>&l | tee m.txt
5 $ make install 2>&l | tee mi.txt
```

El siguiente paso es agregar el programa al **PATH** y podemos comprobar con la instruccion **which**

```
1 $ PATH=/home/pi/mpich-install/bin:$PATH
2 $ echo $PATH
3 /home/pi/mpich-install/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin
  :/usr/bin:/sbin:/bin:/usr/local/games:/usr/games
4 $ which mpicc
5 /home/pi/mpich-install/bin/mpicc
6 $ which mpiexec
7 /home/pi/mpich-install/bin/mpiexec
```

Esto se realiza tanto en el servidor como en cada uno de los clientes, el tiempo de compilacion aproximadamente es de 2 horas.

### 3. Configuración de IP fija

El próximo paso es hacer los ajustes para que tanto el cliente como el servidor puedan compartir datos. Para ambos vamos a configurar que cada uno tenga una IP fija.

```
1 $ sudo nano /etc/network/interfaces
```

En nuestro caso tenemos una red con las direcciones 192.168.169.0/24. Ahora hay que agregar la siguiente línea tanto al servidor como a los clientes.

```
1 auto eth0
2 iface eth0 inet static
3     address 192.168.169.101
4     netmask 255.255.255.0
5     network 192.168.169.0
6     broadcast 192.168.1.255
7     gateway 192.168.169.1
8     dns-nameservers 192.168.169.1
```

La configuración anterior es para el servidor, para los clientes utilizaremos las IP:192.168.169.102 y 192.168.169.103. Ahora lo único que tenemos que hacer es reiniciar para cargar la nueva configuración.

#### 3.1. Servidor

En el servidor podemos editar el archivo **host** para agregar el nombre de los clientes.

```
1 $ sudo nano /etc/hosts
```

En el cual agregamos las siguientes líneas, las cuales corresponden a los clientes que se van a conectar al servidor.

```
1 127.0.1.1      raspberrypi2
2 192.168.169.102 raspberrypi3
3 192.168.169.103 jesus
```

## 4. Llaves SSH

Las maquinas van a comunicarse a través de SSH para compartir información mediante NFS.

Creamos desde el servidor una llave ssh

```
1 ssh-keygen -t rsa -C "pi@raspberrypi"
```

Ahora la copiamos hacia cada de los clientes.

```
1 ssh-copy-id 192.168.169.102
2 /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s
   ), to filter out any that are already installed
3 /usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed — if
   you are prompted now it is to install the new keys
4 pi@raspberrypi3's password:
5
6 Number of key(s) added: 1
7
8 Now try logging into the machine, with: "ssh '192.168.169.102'"
9 and check to make sure that only the key(s) you wanted were added.
```

Para comprobar que todo esta bien configurado nos conectamos desde al cliente sin que nos pida contraseña.

## 5. Configurando NFS

**NFS** es utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras locales. Permite que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. A continuacion se describe los pasos para la configuracion del **servidor** y los **clientes**.

### 5.1. Servidor

Agregar en:

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

La siguiente linea:

```
1 /home/pi 192.168.169.0/24(rw,sync,no_subtree_check)
```

- **192.168.168.0/24:** Indica las direcciones IP a la que compartiremos la carpeta
- **rw:** Habilita la opcion de lectura y escritura (read/write).
- **sync:** Aplica cambios en el directorio compartido, despues de que los cambios son confirmados.
- **no\_subtree\_check:** No hay verificacion del subarbol

### 5.2. Cliente

Agregar en:

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

La siguiente linea:

```
1 192.168.169.101:/home/pi /home/pi nfs defaults 0
0
```

- **192.168.168.101/home/pi:** Indica las direccion IP a la que compartiremos la carpeta.
- **/home/pi:** Indica el punto de montaje.
- **nfs:** Indica el tipo de sistema de archivos.
- **defaults:** No hay verificacion del subarbol.
- **0:** No se toma el cuenta el dispositivo para hace respaldos del sistema de archivos.
- **0:** No se revisa la particion en busca de errores.

## 6. Servicios

El siguiente paso es verificar que los servicios esten levantados, los servicios que estan subrayados son los que se les debe de verificar que tengan un signo + lo cual indica que estan arriba.

### 6.1. Servidor

```
1 $ service --status-all
2 [ - ] mountnfs-bootclean.sh
3 [ - ] mountnfs.sh
4 [ + ] networking
5 [ + ] nfs-common
6 [ + ] nfs-kernel-server
7 [ + ] ntp
8 [ - ] plymouth
9 [ - ] plymouth-log
10 [ + ] procps
11 [ + ] raspi-config
12 [ + ] rc.local
13 [ - ] rmnologin
14 [ + ] rpcbind
15 [ - ] rsync
16 [ + ] rsyslog
17 [ - ] screen-cleanup
18 [ - ] sendsigs
19 [ + ] ssh
20 [ - ] sudo
21 [ + ] triggerhappy
```

### 6.2. Cliente

```
1 $ service --status-all
2 [ + ] networking
3 [ + ] nfs-common
4 [ + ] ntp
5 [ - ] plymouth
6 [ - ] plymouth-log
7 [ + ] procps
8 [ - ] raspi-config
9 [ + ] rc.local
10 [ - ] rmnologin
11 [ + ] rpcbind
12 [ - ] rsync
13 [ + ] rsyslog
14 [ - ] screen-cleanup
15 [ - ] sendsigs
16 [ + ] ssh
17 [ - ] sudo
18 [ + ] triggerhappy
19 [ + ] udev
20 [ + ] udev-finish
21 [ - ] umountfs
```

## 7. Ejecutando programas MPI

Para ejecutar los programas de manera adecuada debemos de iniciar primero el servidor, ya que cada uno de los clientes apuntaran a el para montar el sistema de archivos. Desde los **clientes** podemos verificar con la instruccion **df** si se ha montado de manera adecuada el sistema de archivos.

```
1 $ df
2 Filesystem              1K-blocks    Used Available Use% Mounted
3 /dev/root                15184016 3849880   10662808  27% /
4 devtmpfs                 469544      0     469544   0% /dev
5 tmpfs                   473880      0     473880   0% /dev/shm
6 tmpfs                   473880    6428     467452   2% /run
7 tmpfs                    5120        4        5116   1% /run/lock
8 tmpfs                   473880      0     473880   0% /sys/fs/
9 cgroup
10 /dev/mmcblk0p1          64456    20952     43504   33% /boot
11 192.168.169.101:/home/pi 30361856 3783552  25290496  14% /home/pi
   tmpfs                  94776      0     94776   0% /run/user
   /1000
```

Como se aprecia en la linea 10, se ha montado de manera adecuada.

Ya que todo esta configurado podemos ejecutar el programa que calcula **PI** primero de manera local y luego de manera distribuida. Esto se ejecuta desde el **servidor**.

```
1 $ mpiexec -n 6 ./cpi
2 Process 1 of 6 is on raspberrypi2
3 Process 5 of 6 is on raspberrypi2
4 Process 3 of 6 is on raspberrypi2
5 Process 0 of 6 is on raspberrypi2
6 Process 2 of 6 is on raspberrypi2
7 Process 4 of 6 is on raspberrypi2
8 pi is approximately 3.1415926544231239, Error is 0.0000000008333307
9 wall clock time = 0.031505
10 $ mpiexec -f machines -n 6 ./cpi
11 Process 1 of 6 is on raspberrypi3
12 Process 0 of 6 is on raspberrypi2
13 Process 3 of 6 is on raspberrypi3
14 Process 2 of 6 is on raspberrypi2
15 Process 5 of 6 is on raspberrypi3
16 Process 4 of 6 is on raspberrypi2
17 pi is approximately 3.1415926544231243, Error is 0.0000000008333312
18 wall clock time = 0.002694
```

El archivo **machines** contiene las direcciones IP de los clientes

```
1 $ cat machines
2 192.168.169.101
3 192.168.169.102
```

Se puede apreciar las ventajas del computo distribuido en el tiempo de procesamiento que le toma al cluster el calculo de PI, que es mucho menor que tiempo que le toma a una sola maquina.



## 8. Solucion a problemas

**Problema:** No se puede montar el sistema de archivos en los clientes

Solucion:

Reiniciar en el **servidor** el servicio:

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

y reiniciar todos los clientes.