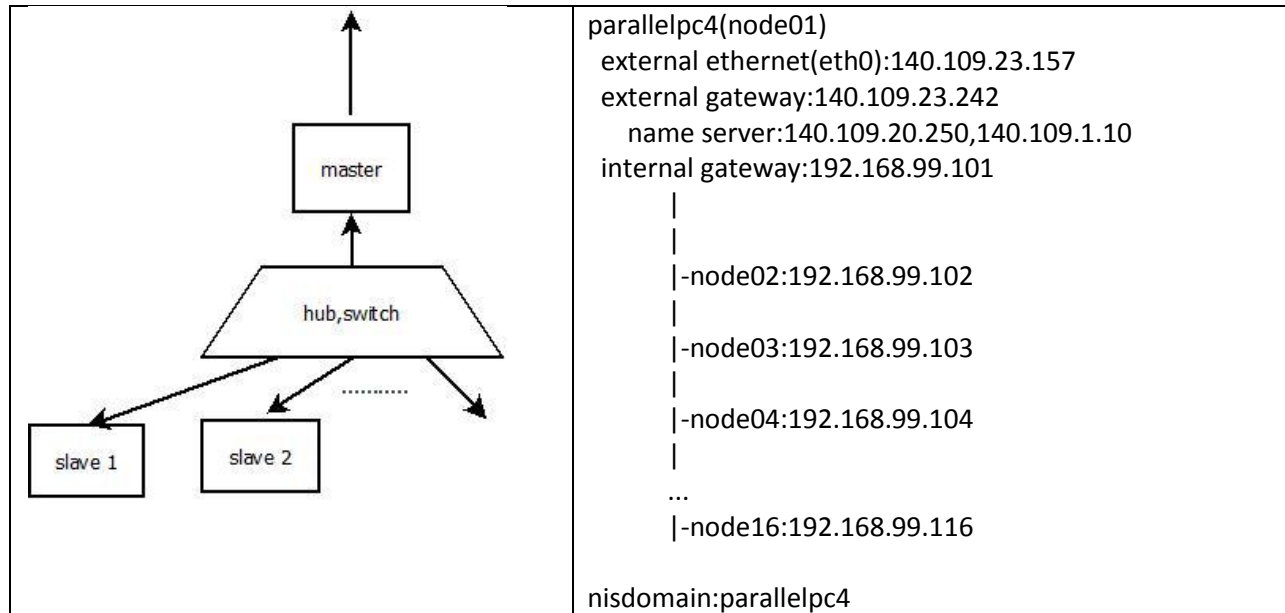


How to set up a cluster computing system

By Tzu-Cheng Chuang



Master node should have 2 ethernet cards.

1. Install CentOS 5.3 64 bit version on all of the machines. Select "Server", "Clustering"

```
# yum update
```

```
# yum install gcc gcc-c++ autoconf automake
```

2. Shutdown Selinux to avoid unexpected warning.

Modify `/etc/sysconfig/selinux`

```
SELINUX=enforcing change this to → SELINUX=disabled
```

3. Change the internet related information.

Master node:

(a) Modify the hostname file `/etc/sysconfig/network`

```
HOSTNAME=hostname.hostdomain
GATEWAY=140.109.100.100
```

(b) Modify the DNS ip address, the file is /etc/resolv.conf

```
nameserver 140.109.200.200
```

(c) Modify /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=140.109.2.255
HWADDR=00:14:5E:30:71:8A
IPADDR=140.109.2.2
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=140.109.2.0
ONBOOT=yes
```

(d) Modify /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=eth1
BOOTPROTO=static
BROADCAST=192.168.99.255
HWADDR=00:0D:60:23:75:EE
IPADDR=192.168.99.101
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=192.168.99.0
ONBOOT=yes
```

Client node:

(a) Modify /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.99.255
HWADDR=00:0D:62:99:75:DE
IPADDR=192.168.99.102
```

```
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.255.0
NETWORK=192.168.99.0
GATEWAY=192.168.99.101
ONBOOT=yes
```

(b) Modify the hostname file /etc/sysconfig/network

```
HOSTNAME=node02
GATEWAY=140.109.99.101
```

(c) Modify the DNS ip address, the file is /etc/resolv.conf

```
nameserver 140.109.200.200
```

4. Edit the host name /etc/hosts

Modify /etc/hosts on every node

```
192.168.99.101 node01 hostname
192.168.99.102 node02
192.168.99.103 node03
.....
```

5. Install NAT service

Master node:

(a) Modify /etc/rc.d/rc.local

After editing this file, reboot the computer.

```
touch /var/lock/subsys/local
echo "1" > /proc/sys/net/ipv4/ip_forward
```

(b) Add the following line in this file /etc/sysconfig/network

```
FORWARD_IPV4=yes
```

(c) Clean IPTALBE RULE (The default rule blocks the client node to resolve the name.)

```
# iptables -F
```

```
# iptables -X
# iptables -Z
# iptables iptables -t nat -A POSTROUTING -o eth0 -s 192.168.99.0/24 -j MASQUERADE
# service iptables save
```

Client node:

(a) Modify /etc/sysconfig/network

```
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=node02
GATEWAY=192.168.99.101
```

(b) Modify /etc/resolv.conf (Change the Name Server)

```
nameserver 140.109.1.1
```

6. Install NFS

Master node:

(a) Start portmap and make it run when reboot :

```
# /etc/rc.d/init.d/portmap start
# chkconfig --level 35 portmap on
```

(b) Set NFS to share the folder :

```
# vim /etc/exports
    /home 192.168.99.0/24(rw,async,no_root_squash)
# exportfs -rv
# /etc/rc.d/init.d/nfs start
# chkconfig --level 35 nfs on
# showmount -e
```

Notice that there is no blank between "/24" and "(rw"

Client node:

(a) Modify /etc/rc.d/local

```
mount -t nfs    node01:/home /home
```

(b) Modify /etc/fstab so that when this machine is reboot, the nfs can be automatically mounted.

```
node01:/home /home nfs auto,hard,bg,intr 0 0
```

(c) Check if the folder is correctly mounted. If it is correct, you should see node01:/home

```
# mount
```

7. Install NIS

Master node:

(a) Install ypserv (NIS server)

```
# yum install ypserv
```

(b) Activate time-dgram and time-stream service

Modify /etc/xinetd.d/time-dgram

```
disable=yes change this to → disable=no
```

Modify /etc/xinetd.d/time-stream

```
disable=yes change this to → disable=no
```

```
# /etc/rc.d/init.d/xinetd restart
```

```
# netstat -utl
```

```
#chkconfig --level 35 time-dgram on
```

```
#chkconfig --level 35 time-stream on
```

(c) Set up NIS domain

```
#nisdomainname cluster
```

```
#echo '/bin/nisdomainname cluster' >> /etc/rc.d/rc.local
```

```
#echo 'NISDOMAIN=cluster' >> /etc/sysconfig/network
```

(d) Modify /etc/ypserv.conf

```
127.0.0.0/255.255.255.0:*:*:none
```

```
192.168.99.0/255.255.255.0:*:*:none
```

```
*:*:*:deny
```

(e) Create /etc/netgroup

```
# touch /etc/netgroup
```

(f) Activate NIS services

```
# /etc/rc.d/init.d/ypserv start
# /etc/rc.d/init.d/yppasswdd start
# /usr/lib64/yp/ypinit -m
# chkconfig --level 35 ypserv on
# chkconfig --level 35 yppasswdd on
```

Every time when the user changes the user information or password, the administrator needs to execute the following command.

```
# /usr/lib64/yp/ypinit -m
```

Client node:

(a) Set up NIS domain

```
#nisdomainname cluster
#echo '/bin/nisdomainname cluster' >> /etc/rc.d/rc.local
#echo 'NISDOMAIN=cluster' >> /etc/sysconfig/network
```

(b) Set up the look up server

Modify /etc/yp.conf

```
domain cluster
ypserver node01
```

(c) Modify the way of password verification

Add the following line in the last part of /etc/passwd

```
+::::::
```

Modify /etc/nsswitch.conf

```
passwd: files nis nisplus
shadow: files nis nisplus
```

group: files nis nisplus

hosts: files nis dns

Modify /etc/sysconfig/authconfig

USENIS=no **change this to** → USENIS=yes

Modify /etc/pam.d/system-auth

password sufficient pam_unix.so md5 shadow nullok try_first_pass use_authtok **Change this to** →

password sufficient pam_unix.so md5 shadow nullok try_first_pass use_authtok nis

(d) Activate NIS

```
# /etc/rc.d/init.d/ypbind start
```

```
# chkconfig --level 35 ypbind on
```

(e) Check if ypbind is running

```
# /usr/sbin/rpcinfo -p localhost
```

or

```
# /usr/sbin/rpcinfo -u localhost ypbind
```

Or

```
# ypcat passwd
```

If there is something coming out, the setting is correct.

8. SSH without password within the cluster

(a) On every node, execute the following.

```
# ssh-keygen
```

(b) Make sure on every node, there is a folder /root/.ssh

Otherwise, create the folder by executing the following.

```
# mkdir ~root/.ssh
```

(c) Copy or append the contents of ~root/.ssh/id_rsa.pub to the other nodes'

```
~root/.ssh/authorized_keys
```

```
# scp -p ~root/.ssh/id_rsa.pub root@node02:~root/.ssh/authorized_keys
```

If the file (~root/.ssh/authorized_keys) on the remote exists already, execute the following

```
# cat ~root/.ssh/id_rsa.pub | ssh root@node02 'cat >> ~root/.ssh/authorized_keys'
```

(d) Copy the finished `/root/.ssh/authorized_keys` to every node.

```
# scp -p ~root/.ssh/authorized_keys root@node03:~root/.ssh/authorized_keys
```

9. Install OPENMPI on every node

(a) Make sure you can compile with GCC. Download `openmpi-1.3.2.tar.gz` to `/home`

(b) Copy `openmpi-1.3.2.tar.gz` to every node `/tmp`

```
# cp /home/openmpi-1.3.2.tar.gz /tmp
```

(c) Unpack `openmpi-1.3.2.tar.gz`

```
# cd /tmp
```

```
# tar xzf openmpi-1.3.2.tar.gz
```

Or

```
# gunzip -c openmpi-1.3.2.tar.gz | tar xf -
```

(e) Configure OPENMPI, specifying the installation directory, and running the configure script in the source directory.

```
# cd /tmp/openmpi-1.3.2
```

```
# ./configure --prefix=/usr/local/openmpi-1.3.2
```

```
# make all install
```

Only on master node,

(f) Modify the `PATH` so that when the system is rebooted, it can run `mpiexec`

Add the following in the file `/etc/skel/.bashrc` and `/etc/skel/.bash_profile` on the master node.

```
export PATH=$PATH:/usr/local/openmpi-1.3.2/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/openmpi-1.3.2/lib
```

Then create a user account for using this cluster.

After compile the file with

```
# mpicc mpi_example.c -o mpi_run
```

Or

```
#mpiCC mpi_example.cpp -o mpi_run
```

The user can run the program with

```
# mpiexec -hostfile hosts.txt -np 4 mpi_run
```