# How to Build an Introductory LINUX Cluster for Parallel Computing

Tao He (tao-he@yuiowa.edu) and Jun Ni, Ph.D. M.E. (jun-ni@uiowa.edu)

Research Services, ITS, the University of Iowa
Iowa City, IA 52242

July 4, 2003

## I. Objective

The objective is to present how to build a simple LINUX cluster for parallel computing. Four major procedures for building such a system follow:

- Design the architecture of LINUX cluster system
- Install RedHat LINUX on each PC (one master or gateway node, and slave nodes), and configure network
- Configure NFS Server on master node and NFS Client on each slave node to for file sharing
- Install and configure Message Passing Library (MPI) on master node

## II. Architecture Design of LINUX Cluster

The basic requirement for building a LINUX cluster is addressed as follows. First, you need to collect multiple PCs and select one PC as a master node and other PCs as slave nodes. The master serves as the gateway to outside network. It requires at least two Ethernet network cards. One card serves as a gateway with a static IP address for outer connection to the Internet. Another serves as a medium for the inter-connection to other slave nodes, through a high-speed switch. The selected PCs for slave nodes should have at least one Ethernet card. Each card connects to the switch through an Ethernet cable.

Only master node is allowed to communicate to the outside network for the reasons of security. The connection between an individual slave node and an outside network is not recommended.

The switch is introduced to group all computer nodes together into a cluster, in which the nodes can communicate with each other during parallel computing. In the present demonstration, we used 3COM-3300 as our inter-connection switch. The switch features the connections with 24 100Mbps ports and possibly packs as a matrix switch system. Basically any fast-speed switch or hub can be served as an internal switch for a LINUX cluster.

The architecture of our simple cluster system is illustrated in the following figure.

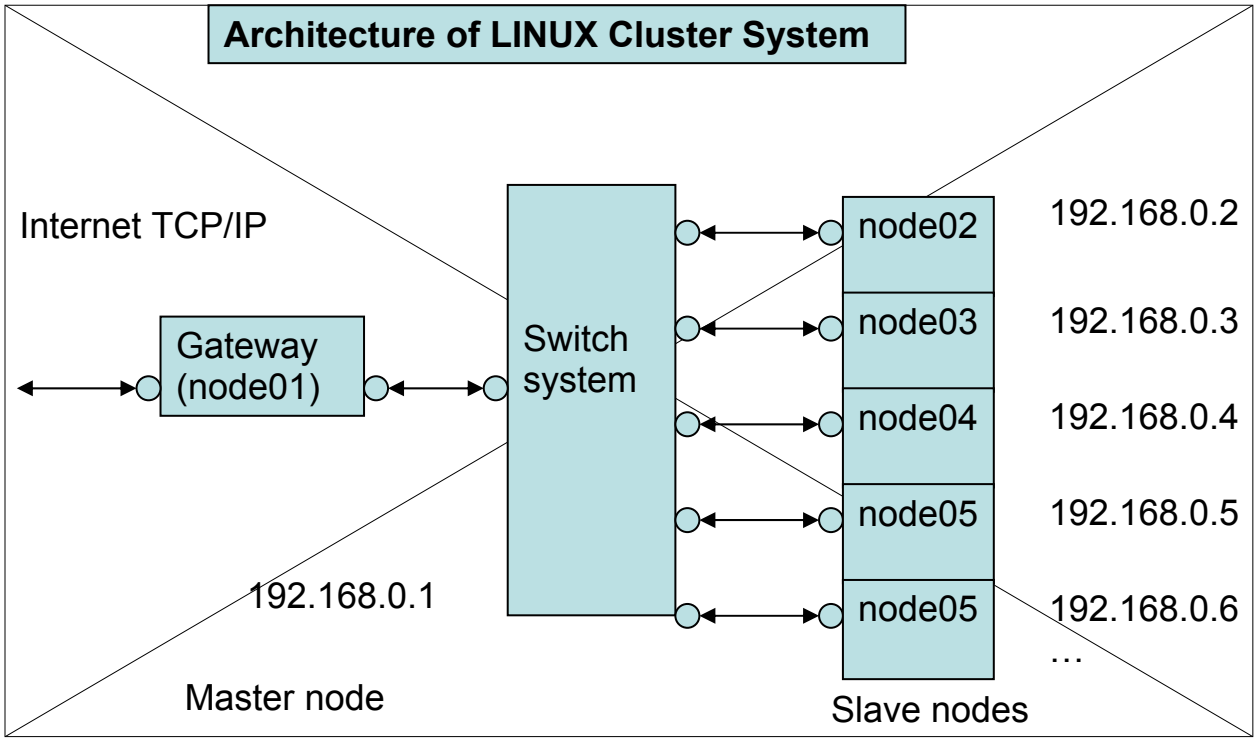


Figure 1 Architecture design of LINUX cluster

**III. Install LINUX and Configure Network**

Theoretically you can select any LINUX operating system. It is totally based on customer's preference. In the present exploration, we selected RedHat 7.3 LINUX OS.

Many methods exist to install LINUX on PCs. One method we used is to boot each PC by using bootable CD of LINUX operating system. The software can be directly downloaded from www.redhat.com. There are three image files you can download from and saved into three CDs, respectively.

First, you install RedHat LINUX on the master node. Then, you can install LINUX on each slave node one by one. Although most of the procedures to install on the master and slave nodes are similar, several special configurations must to be paid attention to.

The step-by-step guidance of installing RedHat LINUX on each PC is listed below.

1. Insert the RedHat LINUX 7.3 CD#1 into the CD-ROM drive of the master node or slave nodes, respectively.

2. Restart or reboot the master node. It should automatically boot your computer. If it doesn't,

you need to check your PC's initial setting (Press **F1** or **Delete** or **F2** key to enter the **BIOS** mode, when you start your PC; Check your computer's specification for details.). Make sure the computer has its initial set-up which allows CD-ROM to boot the system.

3. After booting your PC, the screen will prompt a welcome message. Press the **Enter** key to choose a graphical user interface (GUI) for installation. For the experts of LINUX, it is not necessary to choose the GUI for installation. Instead, you can use **Text** installation mode.

4. Select English as the installation language.

5. Select default keyboard configuration.

6. Select default mouse configuration.

7. Choose the type of installation. RedHat LINUX provides five installation types. They are **Workstation**, **Server**, **Laptop**, **Custom** and **Update**. In the present study, we chose the **Custom** type to install RedHat LINUX on the master and the slave nodes in order to precisely configure the cluster system for better understanding of the installation process.

8. Now you can partition disk space in each node. For the master node, we chose the RedHat's **Disk Druid** partitioning tool. How to partition your hard drive disk totally depends on the customer's opinion. Basically, you should always select the best PC as your master node. We recommend that the master node have a high speed CPU, large memory, and a large disk with high speed. The swap space is usually recommended to configure double the memory size. In the present study, the partition information for our master node is provided in the following for your reference. (Note: since the cluster system for demonstration hereby is a legacy system with recycled PCs, we selected Dell Dimension V400 as our master node.)

| **Partition name** | **Mount point** | **Size** |
|---|---|---|
| root | / | 250M |
| boot | /boot | 50M |
| home | /home | 6G |
| usr | /usr | 2.5G |
| swap | | 128M (2*RAM) |

The partition "**/home**" will be shared with other slave nodes via NFS. The drive "**/usr**" will be used to install MPICH, message passing utility software.

For salve nodes, partition process is very similar to the one used for the master node. If you want to setup the partition quickly, you can choose the default item "**Have the installer automatically partition for you**."

9. Click **Next** button to select the default boot loader "GRUB".

10. Next step is to configure the Ethernet card of master node for networking. RedHat LINUX

will automatically find two Ethernet cards on the master node, named "**eth0**" and "**eth1**," respectively. For configuring "**eth0**" network, we selected the option "**Configure using DHCP**" and "**Activate on boot**". System will automatically generate a TCP/IP address. For configuring "**eth1**" network, we only select the defaulted option "**Activate on boot**". It is time for you to input internal IP address and host name. In the present study, for example, we had

For master node:
IP Address: 192.168.0.1
Host Name: jonathan (you can pick any name)

To configure the Ethernet card for slave nodes, you only select the defaulted option "**Activate on boot**". (Note it is not necessary to select the option "**Configure using DHCP**")

For slave nodes, you can use the following IP mode:
IP Address: 192.168.0.X (i.e., the second node's IP address is 192.168.0.2, and the third is192.168.0.3, …, etc.)
Host Name: node0X (i.e., node02, node03, …, etc.)

11. The next step is to configure firewall security. On the master node we chose the defaulted security level "**medium**". For the trusted device "**eth0**", we selected the items "**DHCP**" and "**SSH**". For the other trusted device "**eth1**", we only selected the items "**SSH**". On slave nodes we only selected the protocol "**SSH**" for the trusted device "**eth0**".

12. In the language support selection, you can choose the language you preferred. We chose "English" item.

13. In the time zone configuration, you can choose time zone of your choice. We chose "American/Chicago: Center Time".

14. Input and confirm the root user's password for future use.

15. In authentication configuration, you can choose the default option.

16. In the package group selection, for the master node you can choose GUI packages, such as "**KDE**" and "**GNOME**". For slave nodes, you don't need to choose GUI and other packages, except the package "**Network support**".

17. It should be noted that you'd activate the checkbox "**Select Individual Packages**", which enables you to install some other protocol package on your cluster system.

18. It is important to note that for all nodes (master and slaves nodes) in our cluster system, we chose the item "**flat view**" to display all package alphabetically. Hereby we chose the following packages, mainly "**openssh,**" "**openssh-server**," "**openssh-client**," "**rsh,**" "**rsh-server**," and "**xinetd.**" After such selections, RedHat LINUX will automatically

play these functions.

19. You have a chance to create a bootable disk (floppy disk) at this time. Alternatively, you do it after the LINUX installation is done. Whenever you need to generate your bootable disk, you can simply use the command "**mkbootdisk**" on console window.

20. Now RedHat LINUX 7.3 installation for a LINUX cluster is completed.

## IV. Network Configuration

Once installation of RedHat LINUX on each node (master & slave nodes), you can configure the network system. Restart your master node. Login on console window, you can use **vi**, or **emacs,** or any editor to modify several configuration files.

- Edit the file **/etc/hosts** for each node (master and slaves) (make sure you log on as a root user). In this study, the following entries are used.

```
127.0.0.1         localhost
192.168.0.1       jonathan
192.168.0.2       node02
192.168.0.3       node03
192.168.0.4       node04
```

- After the complete of the above editing, you should go to the directory "**/etc/init.d**" and enter the command "**./network restart**" to restart the network system.

- In order to test whether your network is successful or not, you can enter the command "**ping <hostname>**", For example

```
$ping node03
PING node03 (192.168.0.3) from 192.168.0.1 : 56(84) bytes of data.
64 bytes from node03 (192.168.0.3): icmp_seq=1 ttl=255 time=1.03 ms
64 bytes from node03 (192.168.0.3): icmp_seq=2 ttl=255 time=0.354 ms
64 bytes from node03 (192.168.0.3): icmp_seq=3 ttl=255 time=0.339 ms
64 bytes from node03 (192.168.0.3): icmp_seq=4 ttl=255 time=0.341 ms
…
4 packets transmitted, 4 received, 0% loss, time 6004ms
rtt min/avg/max/mdev = 0.339/0.443/1.039/0.244 ms

$
```

## V. NFS Server Setup on the Master Node

After configuring the network system of your cluster correctly, you can set up the NFS. Let's show you how we set up NFS Server on our master node. The objective is to configure the master node so that it allows the slave nodes to share its file system.

1. Log on the master node as a "root" user. In the RedHat LINUX console window, enter the command "**setup**". It prompts a setup window. We chose the item "**System Services**" to activate the following daemons: "**network,**" "**nfs,**" "**nfslock,**" "**portmap,**" "**rsh,**" "**rlogin,**" "**sshd,**" and "**xinetd.**" (Note you can select/unselected the item by pressing the **Space** key.)

2. Edit the **/etc/exports** file to specify the file systems (to be shared), hosts (to be allowed) and the type of permissions (ro or rw). In this study, our "**exports**" file has the following entry:

   ```
   /home 192.168.0.1/24(rw, no_root_squash)
   ```

   [Note: 1.) 192.168.0.1/24 presents the first 24 IP addresses, range from 192.168.0.1 to 192.168.0.24 to be allowed to access the exported file system; 2.) No space between 192.168.0.1/24 and (rw, no_root_squash)]

3. Add the new user on the master node. You can use the GUI tool ---"**user manager**" in the KDE or GNOME to add the new user in the system.

4. Reboot the master node (for beginners).

5. If you don't want to reboot at this moment, you can manually restart the NFS service to export the shared directory. First, you need to check the configure file "**/etc/exports**" using the command **"/usr/sbin/exportfs -a**". If there are something wrong in your configure file, it will be reported. Otherwise, you may run the second command "**/sbin/service nfs reload**", which restarts the NFS service and exports the shared directory. As long as you modify the configure file "**/etc/exports**" to change a new shared directory or driver, you need to execute these commands.

**VI. NFS Client Setup on Slave Nodes**

Similarly, we set up the NFS Client on each slave node. The objective is to enable each slave node to mount the file sharing system the master node has.

1. Create the directory "**/home**" on each node (If this directory exist, you don't need to create it.) This directory serves as a mount point of the shared file system established in the master.

2. Log on each slave node as a "root" user. In the console window, enter the command "**setup**", which prompts a setup window. In the setup window, we choose the item "**System Services**" to activate the following daemons: "**network," "nfs,**" "**portmap,**" "**rsh,**" "**rlogin,**" "**sshd,**" and "**xinetd."** (Note you can selected/unselected the item by pressing the **Space** key.)

3. Edit the file "**/etc/fstab**". In this study, we need to append an extra line at the end of the file. This line tells the **home** directory of the master node exactly mounts (links) to the one of current slave node. For example, we had the following line

   ```
   jonathan:/home        /home       nfs
   ```

   This approach enables slave nodes to automatically and statically mount the share

directory of the master node after you reboot your machines.

The alternative, you can use an explicit command "**mount**", which can dynamically mount the master's shared directory to the slave node without rebooting your machines. For example,

```
#mount -t nfs jonathan:/home /home
```

Or you can run the following command to restart NFS client if the file "**/etc/fstab**" is already configured.

```
./netfs restart
```

4. Check whether the NFS is successful or not. We can use the command "**df**" in the LINUX console window. The successful information will display as follows. For example: if you are on node02, you may have the following

```
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda2              3581536    933536   2466064  28% /
/dev/hda1                46636      8846     35382  21% /boot
none                    127956         0    127956   0% /dev/shm
jonathan:/home         3023760    755264   2114896  27% /home
```

5. After the shared directory is setup on the slave node, you are able to add the new user on the slave node. The new user account you add on each slave node should be consistent with the one on the master node. Using the command "**adduser <username>**", and then you can check the "**/etc/passwd**" file to confirm the new user account you had created on the slave node.

6. Reboots this slave node by enter the command "**reboot**".

## VII. Install Parallel Library MPI on Master Node

1. Log on the master node as a root user.

2. Download the MPI parallel library (we chose MPICH version 1.2.5), from the MPI's website (http://www-unix.mcs.anl.gov/mpi/). Save the download of MPICH to the directory called "**/usr**" on the master node. You can create the directory named "**cluster**" under the "**/usr**" directory. Now you can **untar** the package "mpich-1.2.5.tar" in the directory **"/usr/cluster**".

   For example:

   ```
   tar –xvf mpich-1.2.5
   ```

3. Go to the directory "**/usr/cluster/mpich-1.2.5**"

4. Execute the MPI shell file **"./configure**", which checks the environment of the current

computer.

5. Execute the compile command "**make**" to generate the binary codes of MPI.

6. Configure the MPI environment after generating the binary codes. You can go to the directory "**/usr/cluster/mpich-1.2.5/util/machines**". Edit the file "**machines.LINUX**" to specify the computer nodes, which are desirable to join the MPI empowered parallel computing in the cluster system. In this study, we use these entries in the file:

```
jonathan
node02
node03
node04
node05
node06
node07
…
```

## VIII. Configure the cluster system to prepare running the MPI programs

You need to edit the hidden file "**.bash_profile**" to add the path for the MPI. In this study, the path in this file may look like:

```
PATH=$PATH:$HOME/bin:/usr/cluster/mpich-1.2.5/bin:/usr/cluster/mpich-1.2
.5/util
```

## IX. Install and configure the Intel FORTRAN 90 compiler for MPI

After the above setups, you can run C or FORTRAN 77 MPI programs using **"mpicc"** and **"mpif77"** in your cluster**.** To run FORTRAN 90 MPI programs, you need download a FORTRAN 90 compiler and reconfigure MPI for FORTRAN 90. You can download Intel FORTRAN 90 compiler 7.1 and its license file from the Intel website http://www.intel.com/.

You need install FORTRAN 90 compiler package on the master node. In our system, the compiler was installed into the dictionary of **"/home/intel"**. To install Intel FORTRAN 90, you first **untar** the package in a temporary directory. You can put the license file in the default directory

```
/opt/intel/licenses
```

Or you can put the license file in the directory you preferred, such as

```
/home/intel/licenses
```

Then you'd change the install script file **"install"** to reflect the new location of licenses file. After that you need change to root permission and execute **"./install"** to install the compiler. The final step of installation is to update shell environment variables by executing the command

```
#source /<intel_dir>/compiler70/ia32/bin/ifcvars.sh
```

**"<intel_dir>"** is the directory of Intel complier installation. In our example, **"<intel_dir>"** is **"/home/intel"**.

Next step is to build the connection between MPI and the Intel FORTRAN compiler. In our study, we adopted the following steps:

1. Log on the master node with "root" user and change the current directory to the MPI directory. Then execute the environment-setting script **"configure"**

   ```
   #cd /usr/local/mpich-1.2.5
   #./configure –prefix=/usr/cluster/mpich-1.2.5
   --with-common-prefix=/usr/cluster/mpich-1.2.5 –cc=gcc –c++=g++
   -fc=ifc –f90=ifc –enable-sharedlib
   ```

2. Compile the MPI source code.

   #make

3. Update environment variables. In the user shell profile file **".**bash_profile**"**, you should add and export the environment variables **"**LD_LIBRARY_PATH**"** and **"**LIBRARY_PATH**"**

   ```
   LD_LIBRARY_PATH=/<intel_dir>/compiler70/ia32/lib
   LIBRARY_PATH=/usr/cluster/mpich-1.2.5/lib
   export LD_LIBRARY_PATH LIBRARY_PATH
   ```

## X. Test MPI program

After you successfully install MPI, you should be able to test your parallel code enhanced by MPI library.

```
$mpicc –o MPIcode.o MPIcode.c
$mpirun –np 10 MPIcode.o
```

or

```
$mpicc MPIcode.c
$mpirun –np 10 a.out
```

Congratulation! A simple but useful LINUX cluster is in you hand.

# How to build the master node for a simple LINUX cluster

## I. Installs the LINUX on our master computer

The step-by-step guidance of installing LINUX on master PC is listed below.

1. Insert the RedHat LINUX 7.3 CD#1 into the CD-ROM drive of the master node.

2. Restart or reboot the master node. It should automatically boot your computer. If it doesn't, you need to check your PC's initial setting (Press **F1** or **Delete** key, or F2 key to enter the **BIOS** mode, when you start your PC; check your computer's specification for details). Make sure the computer has its initial set-up, which allows CD-ROM to boot the system.

3. After booting your PC, the screen will prompts a welcome message. Press the **Enter** key to choose a graphical user interface (GUI) for installation. For the experts of LINUX, it is not necessary to choose the GUI for installation. Instead, you can use **Text** installation mode.

4. Select English as the installation language.

5. Select default keyboard configuration.

6. Select default mouse configuration.

7. Choose the type of installation. RedHat LINUX provides five installation types. They are **Workstation**, **Server**, **Laptop**, **Custom** and **Update**. In the present study, we chose the **Custom** type to install RedHat LINUX on the master node and the slave nodes, in order to precisely configure our cluster system for better understanding of the installation process.

8. Now you can partition disk space in each node. For the master node, we chose the RedHat's **Disk Druid** partitioning tool. How to partition your hard drive disk totally depends on the customer's opinion. Basically, you should always select the best PC as your master node. We recommend that the master node have a high speed CPU, large memory, and a large disk space. The swap space is usually recommended to configure double the memory space built-in. In the present study, the partition information for our master node is provided in the following for your reference. (Note: since the cluster system for demonstration hereby is a legacy system, we selected Dell Dimension V400 as our master node.)

| **Partition name** | **Mount point** | **Size** |
|---|---|---|
| root | / | 250M |
| boot | /boot | 50M |
| home | /home | 6G |
| usr | /usr | 2.5G |
| swap | | 128M (2*RAM) |

The drive "**/home**" will be shared with other slave nodes via NFS. The drive "**/usr**" will be used to install MPICH, message passing utility software.

9. Click **Next** button to select the default boot loader "GRUB".

10. Next step is to configure our Ethernet card of master node for networking. RedHat LINUX will automatically find two Ethernet cards on the master node, named "**eth0**" and "**eth1**," respectively. For configuring "**eth0**" network, we selected the option "**Configure using DHCP**" and "**Activate on boot**." For configuring "**eth1**" network, we only select the defaulted option "**Activate on boot**." It is time for you to input internal IP Address and host name. In the present study, for example, we had

    For master node:
    IP Address: 192.168.0.1
    Host Name: jonathan (you can pick any name)

11. The next step is to configure firewall security. On the master node we chose the defaulted security level "**medium**". For the trusted device "**eth0**," we selected the items "**DHCP**" and "**SSH**." For the other trusted device "**eth1**," we only selected the items "**SSH**."

12. In the language support selection, we chose "English" item.

13. In the time zone configuration, we chose "American/Chicago: Center Time".

14. Input and confirm the root user's password for future use.

15. In authentication configuration, we chose the default option.

16. In the package group selection, for the master node we chose "**KDE**" and "**GNOME**", respectively.

17. It should be noted that activate the checkbox "**Select individual Packages**", which enable us to install some other protocol package on our cluster system.

18. It is important to note that for all nodes (master and slaves nodes) in our cluster system, we chose the item "**flat view**" to display all package alphabetically. Hereby we chose the following packages, mainly "**openssh,**" "**openssh-server**," "**openssh-client**," "**rsh,**" "**rsh-server**," and "**xinetd.**" After such selections, RedHat LINUX will automatically play these functions.

19. You have a chance to create a bootable disk (floppy disk) at this time. Alternatively, you do it after the LINUX installation is done. Whenever you need to generate your bootable disk,

you can simply use the command "**mkbootdisk**" on console window.

20. Now RedHat LINUX 7.3 installation for a LINUX cluster is completed.

21. Logs in the master node with the user "root" and edits the file **/etc/hosts** for each node (master and slaves) (logging in with root user). In this study, the following entries are used.

```
127.0.0.1      localhost
192.168.0.1    jonathan
192.168.0.2    node02
192.168.0.3    node03
192.168.0.4    node04
```

## II. Configures the NFS server

1. Log in the master node as a "root" user. In the LINUX console window, enter the command "**setup**". In the setup window LINUX provided, we choose "**System Services**" to active the following daemons: "**network**", "**nfs**", "**nfslock**", "**portmap**", "**rsh**", "**rlogin**", "**sshd**", and "**xinetd**". Note we select the item or unselect the item by the key of "Space".

2. Edit the **/etc/exports** file to specify the file systems (to be shared), **hosts** (to be allowed) and the type of permissions (ro, rw). In this study, our "**exports**" file has the following entry:

```
/home  192.168.0.1/24(rw, no_root_squash)
```

Note: (1) 192.168.0.1/24 presents the first 24 IP addresses, from 192.168.0.1 to 192.168.0.24 to access the exported filesystem.
(2) No space between 192.168.0.1/24 and (rw, no_root_squash)

3. Add the new user on the master node. Using the GUI tool ---"**user manager**" in the KDE or GNOME to add the new user in our system.
4. Reboot the master node

## III. Installs the parallel software---MPICH version 1.25

4. Logs in the master node as a root user

5. Download the MPI parallel library, MPICH version. 1.2.5, from the MPI's website. Save the download of MPICH in the directory called "**/usr**" on the master node. You can create the directory named "**cluster**" in the "**/usr**". Now you can **untar** the package "mpich-1.2.5.tar" in the directory **"/usr/cluster**".

For example:

```
tar –xvf mpich-1.2.5
```

6. Go to the directory "**/usr/cluster/mpich-1.2.5**"

7. Runs the mpi shell file **"./configure**", which will check the environment of the current computer.

8. Runs the compile command "**make**" to generate the binary files of MPI.

9. Configures the MPI environment after generate these binary files. You can go to the directory "**/usr/cluster/mpich-1.2.5/util/machines**". Edit the file "**machines.LINUX**" to specify the computer nodes, which are desirable to join the MPI cluster computing. In this study, we use these entries in the file:

   jonathan
   node02
   node03
   node04

## How to build the slave node for a LINUX cluster

### I. Installs the LINUX on our every slave computer

1. Insert the RedHat LINUX 7.3 CD#1 into the CD-ROM drive of every slave PC.

2. Restart or reboot the master node (computer). It should automatically boot your computer. If it doesn't boot, you need to check your PC's initial setting. Make sure the computer has set-up, which allows CD-Rom to boot the system.

3. After booting, the screen will display a welcome message. Press the **Enter** key to choose the graphical user interface (GUI) to install. For experts of LINUX, it is not necessary to choose GUI to install. You can use **Text** interface.

4. Select English as the installation language.

5. Select default keyboard configuration.

6. Select default mouse configuration.

7. Choose the type of installation. RedHat LINUX provides five installation types: **Workstation**, **Server**, **Laptop**, **Custom** and **Update**. In the present study, we chose the **Custom** type to install LINUX on the master node and the slave node, in order to precisely configure our system for better understanding of the installation process.

8. Setup Disk Partition. For the master node we chose the Disk Druid partitioning tool, provided by LINUX operating system. For salve node, the partition process is very

similar to the one used for master node. It is still able to be used in the slave node. The disk partition totally depends on the customer's opinion. Basically, you should always select the best PC as the master node. We recommend that the master node have high speed CPU, large memory, and large disk space. The swap space is usually recommended to configure double the memory space. In the present study, the partition information for the master node we used is provided in the following for your reference. For example:

| partition name | mount point | size |
|---|---|---|
| root | / | 2.5G |
| boot | /boot | 50M |
| swap | | 128M (2*RAM) |

If you don't mind and want to fast setup the partition table, you could choose the item "Have the installer automatically partition for you."

9. Select the default boot loader "GRUB".

10. Next step: to configure our Ethernet card of slave node for networking. LINUX will automatically find the Ethernet card in the slave node, named "eth0". For configuring "eth0" network, we deselected the option "Configure using DHCP", but we need to select "Activate on boot." In the present study, we use

For slave node
IP Address: 192.168.0.X
(For example, the second node's IP address is 192.168.0.2, and the third node's IP address is192.168.0.3, etc.)
Host Name: node0X (for example: node02, node03, etc.)

11. The next step is to configure firewall security. On the master node we chose the security level "**medium**". On the slave node we just select the protocol "SSH" for the trusted device "eth0".

12. In the language support selection, we chose "English" item.

13. In the time zone configuration, we chose "American/Chicago: Center Time".

14. Input the root user's password. It is time for you to choose root password.

15. In authentication configuration, we chose the default option.

16. In the package group selection, For the slave nodes, since it is not necessary to have GUI, we didn't choose GUI packages and other packages except the package "Network support".

17. Then we must remember to select the checkbox “Select individual Packages”, because we need to install some other protocol package into our cluster system.

18. It is important to note that for all nodes (master and slaves nodes) in our cluster system, we chose the item “flat view” to display all package according to the sort of alphabet. Here we chose the following packages, including "**openssh,**" “**openssh-server**,” “**openssh-client**,” "**rsh,**" “**rsh-server**,” "**xinetd.**" After the selections, LINUX will automatically play these functions on your computer.

19. After installation, you have a chance to create a boot disk. You can create bootable disk (floppy disk) at this time. However, you may also create your bootable disk whenever you want. You can use the command "**mkbootdisk**" after LINUX has been setup.

20. Now RedHat LINUX 7.3 installation is completed.

21. Logs in the slave node with the user “root” and edits the file **/etc/hosts** for each node (master and slaves) (logging in with root user). In this study, the following entries are used.

```
127.0.0.2      localhost
192.168.0.5    jonathan
192.168.0.6    node02
192.168.0.7    node03
192.168.0.8    node04
```

Note: If a new slave node is added into the system, we need to modify the file “/etc/hosts” on all PCs of the system to reflect the IP address of the adding slave node.

## II. Configures the NFS Client

1. Logs in the slave node as a "root" user. In the LINUX console window, enter the command "**setup**". In the setup window LINUX provided, we choose "**System Services**" to active the following daemons: "**network**", "**nfs**", "**portmap**", "**rsh**", "**rlogin**", "**sshd**", and "**xinetd**". Note we select the item or unselect the item by the key of “Space”.
2. Creates the directory "**/home**" on each node (If this directory is not existing in the slave node. We need the directory to work as the mount point of remote file system)
3. Edits the file "**/etc/fstab**". In this study, we add a line on the bottom of this file to tell the current slave node the information about the remote host name, the shared directory, and the local mount directory.

```
jonathan:/home      /home      nfs
```

This approach enables slave nodes to automatically and statically mount the share directory of a remote node, whenever you reboot your machines.

The alternative method is to type the command "**mount**", which can be used to dynamically mount to the remote share directory without rebooting your machines.

For example:

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

4. Checks whether the NFS runs successfully or not. We use the command "**df**" in the LINUX console window. The successful information will display as following: (for example: node02)

| Filesystem | 1k-blocks | Used | Available | Use% | Mounted |
|---|---|---|---|---|---|
| /dev/hda2 | 3581536 | 933896 | 2465704 | 28% | / |
| /dev/hda1 | 46636 | 8846 | 35382 | 21% | /boot |
| jonathan:/home | 3023760 | 39824 | 2830336 | 2% | /home |

5. After the shared directory had setup in the slave node, you are able to add the new user on the slave node. Note the new user should refer to the master node, in the other word, you should add the same user with the master node, especially the same sort. Using the command "**adduser <username>**", and then we check the "**/etc/passwd**" file to confirm the new user account you just created has been added in the slave node.

6. Reboots this slave node.

Congratulation! A simple and useful LINUX Cluster has been in your hand.