Linux Clustering

What is Linux :

. Linux is an open-source Unix like operating system. Linux has a reputation of a very secure and efficient system. It is used most commonly to run network servers and has also recently started to make inroads into Microsoft dominant desktop business. It is available for wide variety of computing devices from embedded systems to huge multiprocessors, also it is available for different processors like x86, powerpc, ARM, Alpha, Sparc, MIPS, etc.It should be remembered that Linux is essentially the OS Kernel developed by **Linus Torvald** and is different from the commonly available distributions like RedHat, Caldera,etc(These are Linux Kernel plus GPLed softwares).

Common Commands in Linux:

1.1 Changing directory

cd without arguments puts the user in the users home directory. With a directory name as argument, the command moves the user to that directory

$>cd ***directorypath***

1.2 Copy files

cp makes copies of files in two ways.

$>cp ***file1 file2***

makes a new copy of **file1** and names it **file2**.

$>cp ***[list of files] directory***

puts copies of all the files listed into the directory named. Contrast this to the **mv** command which moves or renames a file.

1.3 Making a link

ln creates a link between files.
Example:
The following links the existing file example.c to ex.c.

$>ln ***example.c ex.c***

The following creates symbolic links.

$>ln -s ***/usr/include incl***

See the online **man** pages for many other ways to use **ln**.

1.4 Make a new directory

mkdir makes a new subdirectory in the current directory.

$>mkdir *directoryname*

makes a subdirectory called **directoryname**.

1.5 Move / rename files

mv moves or changes the name of a file.

$>mv *file1 file2*

changes the name of **file1** to **file2.** If the second argument is a directory, the file is moved to that directory. One can also specify that the file have a new name in the directory ‘direc’:

$>mv *file1 direc/file2*

would move **file1** to directory **direc** and give it the name **file2** in that directory.

1.6 Present working directory

pwd returns the name of the current working directory. It simply tells you the current directory.

1.7 Remove files

rm removes each file in a list from a directory. By default option **-i** to **rm** inquires whether each file should be removed or not. Option **-r** causes rm to delete a directory along with any files or directories in it.

$>rm *filename*

1.8 Remove directory

rmdir removes an empty directory from the current directory.

$>rmdir *directoryname*
removes the subdirectory named **directoryname** (if it is empty of files). To remove a directory and all files in that directory, either remove the files first and then remove the directory or use the **rm –r** option described above.

1.9 Listing files and directories

ls lists the files in the current directory or the directory named as an argument. There are many options:

ls -a [directory]
lists all files, including files whose names start with a period.

ls -c [directory]
lists files by date of creation.

ls -l [directory]
lists files in long form: links, owner, size, date and time of last change.

ls -p [directory]
subdirectories are indicated by /.

ls -r [directory]
reverses the listing order.

l**s -s [directory]**
gives the sizes of files in blocks.

ls -C [directory]
lists files in columns using full screen width.

ls -R [directory]
recursively lists files in the current directory and all subdirectories.

2 File Transfer

2.1 Establishing remote connection

To establish a connection to a remote system use the **sftp** command. After the connection is established provide the valid password.

```
$>sftp –oPort=44 user@203.90.127.210
```

2.2 File uploading

Move a file from the local host to remote host

```
$>put filename
```

To put multiple files using wild cards

```
$>mput pattern*
```

2.3 File downloading

Move a file from remote host to local host

```
$>get filename
```

To get multiple files using wild cards

```
$>mget pattern*
```

2.4 Making a new directory on remote host

```
$>mkdir directoryname
```

2.5 Changing directory in local host

```
$>lcd directorypath
```

2.6 Changing directory in the remote host

```
$>cd directorypath
```

2.7 Closing the connection

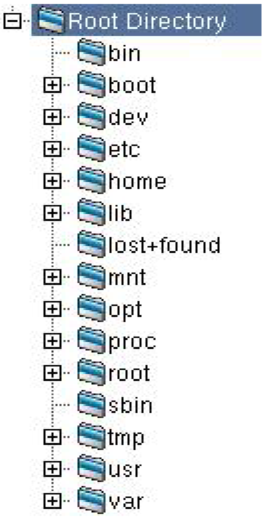
```
$>bye
```

File structure in Linux:

Data and programs are stored in **files,** which are segmented in directories. In a simple way, a directory is just a file that contains other files (or directories). The part of the hard disk where one is authorized to save data is called **home directory**. Normally all the data that is to be save will be saved in files and directories in the home directory. The symbol ~ can also be used for home directory.

The directory structure of **Linux** is a tree with directories inside directories, several levels .The tree starts at what is called the **root directory** **/** (slash).

The following are the list of directories or say branches of the tree.



1. **/bin**: contains basic utilities like bash,chmod,chown,date,df,kill,mkdir,mount etc
2. **/boot**: a copy of the kernel (Linux) needed for the machine to start up (to boot).
3. **/cdrom**: to read CDs .
4. **/dev**: in Linux every hardware is essentially a file which resides here.
5. **/etc**: the system configuration files and directories like bashrc, init.d, profile.d, yp.conf of the system.
6. **/floppy**: to read floppies.
7. **/home**: typically has the user directories to store personal files.
8. **/initrd**: another set of files needed for the machine to boot.
9. **/lib**: files (called libraries) needed for programs to work.
10. **/mnt**: a directory for temporarily reading some hardware devices, mount points for temporary mounts by the system administrator.
11. **/proc**: a virtual directory created by the currently running kernel to store information about all the running system/user processes. It is deleted when the system is shut down.
12. **/sbin**: these files are utility files used for system management .

13. **/usr**: a (huge) directory with many programs. The /usr directory is designed to store static, sharable, read-only data. Programs which are used by all users are frequently stored here. Data which results from these programs is usually stored elsewhere.
14. **/root**: the directory where the system administrator (root) saves his/her files
15. **/tmp**: a temporary directory used by many programs to save things for short periods of time (files here are periodically removed).
16. **/var**: contains variable data, mostly stuff needed for the system to work (like PID information) or databases. This directory stores **var**iable data like logs, mail, and process specific files. Most, but not all, subdirectories and files in the /var directory are shared.

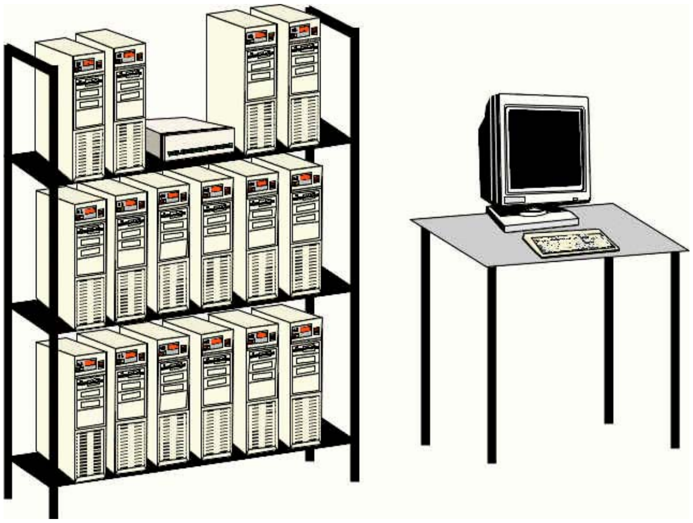
What is clustering & why it is required ?

clustering is the use of multiple computers, typically PCs or UNIX workstations, multiple storage devices, and redundant interconnections, to form what appears to users as a single highly available system. Cluster computing can be used for load balancing as well as for high availability. Cluster computing is used as a relatively low-cost form of parallel processing machine for scientific and other applications that lend themselves to parallel operations.

Computer cluster technology puts clusters of systems together to provide better system reliability and performance. Cluster server systems connect a group of servers together in order to jointly provide processing service for the clients in the network.

Cluster operating systems divide the tasks amongst the available servers. Clusters of systems or workstations, on the other hand, connect a group of systems together to jointly share a critically demanding computational task. Theoretically, a cluster operating system should provide seamless optimization in every case.

At the present time, cluster server and workstation systems are mostly used in High Availability applications and in scientific applications such as numerical computations.



.

. **Clusters can offer**

. • High performance

. • Large capacity

. • High availability

. • Incremental growth

. • **Clusters used for**

. • Scientific computing

. • Making movies

. • Commercial servers (web/database/etc)

Requirements

The main requirements that a clustering algorithm should satisfy are:

. • **scalability;**

. • **dealing with different types of attributes;**

. • **discovering clusters with arbitrary shape;**

. • **minimal requirements for domain knowledge to determine input parameters;**

. • **ability to deal with noise and outliers;**

. • **insensitivity to order of input records;**

. • **high dimensionality;**

. • **interpretability and usability.**

Why Linux is used in cluster building?

There are some issues which give unmountable advantages to Linux for the purpose.

- Linux runs on a wide range of hardware
- Linux is exceptionally stable
- Linux source code is freely distributed.
- Linux is relatively virus free.
- Having a wide variety of tools and applications for free.

GETTING STARTED WITH LINUX CLUSTER:

Introduction:

Cluster computing is a very economical form of parallel computing. The configuration that is described in this document is based on the concept of a Beowulf cluster, using publicly available OSCAR software package. We followed these steps to build our cluster we expect them to be complete and totally safe through the instructions are without any safety guarantee whatsoever.

Keywords used in the clustering process & details :

- Installation
- Networking
- File structure of Linux
- NFS
- NIS
- RPM
- PVM
- PBS
- IP address
- MPI
- Parallel Programming
- Parallel software Libraries
- Open source package eg: OSCAR.
- SSH
- SGE
- Autofs
- Display export

Network File System (NFS):

A distributed file system that enables users to access files and directories located on remote computers and treat those files and directories as if they were local. NFS is independent of machine types, operating systems, and network architectures through the use of remote procedure calls (RPC).

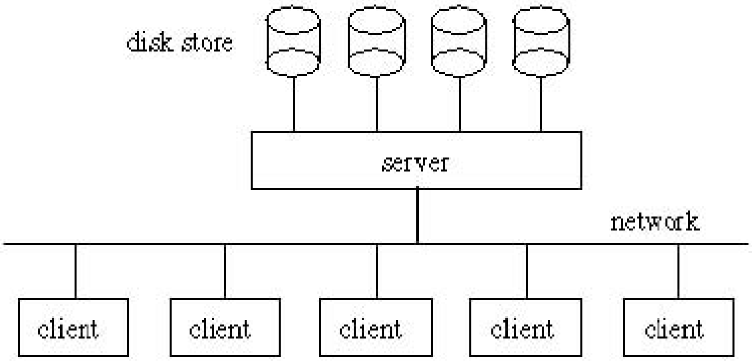


Figure: The division of NFS between client and server

Network Information System (NIS):

Network Information Service, a service that provides information , that has to be known throughout the network, to all machines on the network. NIS is a distributed database that provides a repository for storing information about hosts, users, and mailboxes in the UNIX environment. It was originally developed by Sun Microsystems and called YP (Yellow Pages). NIS is used to identify and locate objects and resources that are accessible on a network.

RedHat Package Manager (RPM):

Linux files are generally RPMs. RPM also stands for Red Hat Package Manager. Red Hat Linux uses the **RPM** technology of software installation and upgrades. Using **RPM**, either from the shell prompt or through **Gnome-RPM**, is a safe and convenient way to upgrade or install software.

Parallel Virtual Machine (PVM):

PVM (Parallel Virtual Machine) is a software package that permits a heterogeneous collection of Unix and/or Windows computers hooked together by a network to be used as a single large parallel computer. The individual computers may be shared- or local-memory multiprocessors, vector supercomputers, specialized graphics engines ,or scalar workstations, that may be interconnected by a variety of networks ,such as ethernet , FDDI.

Portable Batch System (PBS):

OpenPBS is the original version of the Portable Batch System. It is required for the scheduling of the jobs.It is a flexible batch queueing system. It operates on networked, multi-platform UNIX environments. OpenPBS consists of three primary components—a job server(pbs_server) handling basic queuing services such as creating and modifying a batch job and placing a job into execution when it's scheduled to be run. The executor (pbs_mom) is the daemon that actually runs jobs. The job scheduler (pbs_sched) is another daemon. pbs_server and pbs_sched are run only on the front end node(server node), while pbs_mom is run on every node of the cluster that can run jobs,including the front end node(server).

Message Passing Interface (MPI):

A widely accepted standard for communication among nodes that run a parallel program on a distributed-memory system . The standard defines the interface for a set of functions that can be used to pass messages between processes on the same computer or on different computers. MPI can be used to program shared memory or distributed memory computers. Hence MPI is a library of routines that can be called from Fortran and C programs There are a large number of implementations of MPI, two open-source versions are MPICH and LAM.

Secure Shell (SSH):

A packet-based binary protocol that provides encrypted connections to remote hosts or servers. Secure Shell is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. It is a replacement for rlogin, rsh, rcp, and rdist ,telnet ,ftp.

Cluster Components:

The cluster consists of four major parts. These parts are: 1) Network, 2) Compute nodes, 3) Master server, 4) Gateway. Each part has a specific function,that are needed for the hardware to perform its function.

1. **Network:**

    - Provides communication between nodes, server, and gateway
    - Consists of fast ethernet switch, cables, and other networking hardware

2. **Nodes:**

- Serve as processors for the cluster
- Each node is interchangeable, there are no functionality differences between nodes
- Consists of all computers in the cluster other than the gateway and server

3. **Server:**

- Provides network services to the cluster
- DHCP
- NFS (Node image and shared file system)
- Actually runs parallel programs and spawns processes on the nodes
- Should have minimum requirement.

4.Gateway:

- Acts as a bridge/firewall between outside world and cluster
- Should have two ethernet cards

Hardware needed:

General Hardware:

- Requirements
  - Ethernet switch
  - Monitor switch for computers
  - Power strips
- Optional
  - Backup power supply
  - Racks for computers

Software needed:

In order to make your cluster run you will need several software packages. These are any Linux distribution we used RedHat Enterprise LINUX 3 (your operating system), ssh (your communications package),LAM/ MPICH (the software that allows you to run parallel programs), and ntp timeserver (not essential, however, it does help keep the time on the cluster synchronized).

Why Synchronize Your Network Time?

Synchronized time is critical for the enterprise network because:

- To reduce confusion in shared filesystems it is crucial for the timestamps to be consistent.
- To manage, secure and debug your network you need to accurately know when events happen.

- By installing a time server within your firewall, risks from the outside are minimized and the timing accuracy on your network is maximized.

Cluster administration

Cluster requires maintenance and management to achieve the best performance.

Topics Included

. • **User account management**
. • **Cluster network security**
. • **Software installation and maintenance**
. • **Backup Startegies**
. • **Cluster monitoring**
. • **Performance testing**
. • **Nfs**
. • **LAM**
. • **Configuring rsh**
. • **Configuring ssh**
. • **MPICH overview**
. • **PBS overview**

First start with a cluster of two Computers using LAM(Local Area multicomputer)

LAM/MPI is a high-quality open source implementation of the Message Passing Interface specification. LAM/MPI contains a wide variety of features for system administrators ,parallel programmers, and application users.

First of all download LAM files and RPM's from the site and install it in a directory.

This directory is your working directory. Create a hostfile which provide a listing of the machines to be included in an MPI session .Here we are taking two machine where as you can take any number of machines.

Make users on the two machines (you can take any name).

Let us assume the names of two machines are Linster1 and Linster2.

Here is an example of hostfile

```
  Linster1

 Linster2
```

You should be sure that for each of the hosts listed in the hostfile there is a corresponding entry for that host in your .rhosts file.

Your .rhosts file should looks like in the following format

```
Linster1 user=username(which you have set on first machine)

Linster2 user=username(which you have set on second machine)
```

Please note that the setup which we described should supports a homogeneous environment ,ie ; the should be of same built and running same operating system say Linux9.

Now intial step to make the setup ready is over.

Now come to booting LAM

lamboot

TO initiate a MPI session under LAM just type at the unix prompt

```
$ lamboot –v hostfile
```

The –v is given so that it shows how LAM actually do the operations step by step.

At the end it shows **‘topology done’** on the screen.

This command will execute boot on different node.

If you want to check whether the MPI session is running uninterruptly just type ,**tping –c3 N** on the shell prompt. N command line argument is to tell LAM to ping all the host in the hostfile where as –c3 specify a number of ping statement to execute.

Once this is over we have to compile the MPI programs.

Compiling MPI Programs

mpicc ,mpiCC (or mpic++),and mpif77 are the wrapper compilers for C, C++, and fortran programs respectively.

These compilers includes all the relevant files and directories required for running the MPI programs.

For an example to compile a program hello world in ‘C’

```
Shell$ mpicc hello.c –o hello
```

Once the LAM universe is established successfully and the MPI program is compiled, MPI program can be run in parallel.

Executing MPI Programs

A MPI application is started by one invocation of the mpirun command.

```
Shell$ mpirun –v np 2 hello  (where np=number of processors)
```

It means one copy of program is launched on n0(named node1) and other copy of program is launched on other n1 (named node2).

Shutting down LAM

After the work is over and LAM/MPI no longer needed the MPI session must be shutdown using the command

```
$lamhalt
```

If lamhalt hangs and does not return to command prompt ,use the command

```
$wipe –v hostfile
```

This will kill all the hosts in your hostfile.

It is essentially important that the MPI sessions must be terminated ,otherwise it will cause unpredictable results or even the machine crash could happens.

Type of clusters :

- Failover Clusters.
- Scalable High Performance Clusters.
- Application Clusters.
- Network Load balancing clusters
- Other types of clusters.

Here we are using OSCAR (open source cluster Application Resource) software package, which is a high performance (HPC) cluster and is freely available open source software package.

What is OSCAR?

The OSCAR software package is used to simplify the complex tasks required to install a cluster .The benefit for using this package is that several HPC-related packages are installed by default and we need not to install them separately like MPI implementations LAM , PVM ,PBS ,etc.

Supported Distributions

The following is a list of supported Linux distributions for the OSCAR-4.1 package:

- Red Hat Linux 9 (x86)
- Red Hat Enterprise Linux 3 (x86, ia64)
- Fedora Core 3 (x86)
- Fedora Core 2 (x86)
- Mandriva Linux 10.0 (x86)

Each individual machine of a cluster is reffered to as a node.In OSCAR cluster there are two types of nodes: server and client.

A server is responsible for serving the requests of client nodes,whereas a client is dedicated to computation.

An OSCAR cluster consists of one server node and one or more client nodes,where all the client nodes must have homogeneous hardware.

The Cluster which we are having is named as **Linster,** which is originated from the term **Lin**ux Clus**ter**.

Configuration of **Linster:**

- 16 node cluster.
- 1 Server node and 15 Client nodes.

SERVER:

- 1.5 GHz Pentium P4 Processor.
- Two 40 GB Hard disks.
- 256 MB RAM.
- Two network interface cards supporting TCP/IP stack.
- Keyboard and mouse.

CLIENTS:

- 1.5 GHz Pentium P4 Processor.
- One 40 GB Hard disk.
- 256 MB RAM.
- One network interface card supporting TCP/IP stack.
- No keyboard or mouse is required.

The main advantage of using OSCAR is that we need not to do any operation on client nodes all the process has to be done on the server node starting from the linux installation to cluster setup. The client’s settings can be taken care of by the OSCAR package itself .All we have to do is to network boot the clients and it will automatically copy the images from the Server .

Linster’s public IP address is **10.96.6.25.**

Setting up the Cluster:

Steps involved in installing the OSCAR distribution on the sever

- Install RedHat linux on the Server node and make partition accordingly.(We are using RedHat Linux Enterprise 3).
- Download OSCAR distribution package from http://oscar.sourceforge.net/
- Go to the OSCAR directory by using command

```
# cd  /root/oscar-4.1
```

then run the configure script

```
# ./configure.
```

Now you are ready to actually install OSCAR on the server.

Set environment variables like OSCAR_HOME .

Run #make install

The default directory is /opt/oscar in which oscar is to be installed.

- Now copy distribution installation RPMs to /tftpboot/rpm directory from linux distribution CD by using the command

```
#cp /mnt/cdrom/RedHat/RPMS/*.rpm   /tftpboot/rpm.
```

If any rpm is missing then download it from the site,

ftp://ftp.redhat.com/pub/redhat/linux/enterprise/3/en/RPMS

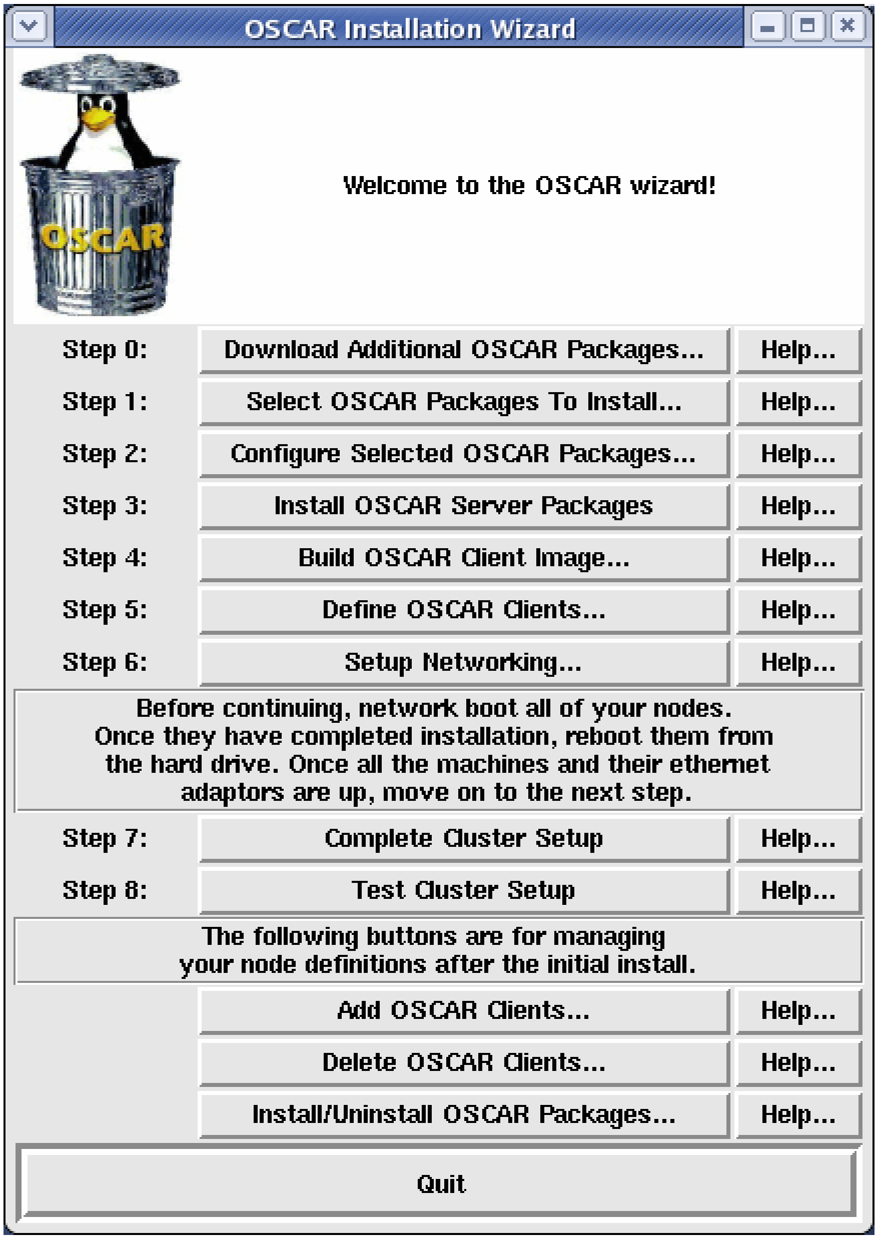
- Change directory to top level OSCAR directory

```
#cd /$OSCAR_HOME

# ./install_cluster <device>
```

substitute the device name (e.g.,eth1) for server's private network ethernet adapter.While running this command a OSCAR installation wizard will appear on the screen.

Run all the steps which are given in the wizard according to their instructions.



Welcome to the OSCAR wizard!

Step 0: Download Additional OSCAR Packages...

Step 1: Select OSCAR Packages To Install...

Step 2: Configure Selected OSCAR Packages...

Step 3: Install OSCAR Server Packages

Step 4: Build OSCAR Client Image...

Step 5: Define OSCAR Clients...

Step 6: Setup Networking...

Before continuing, network boot all of your nodes.
Once they have completed installation, reboot them from
the hard drive. Once all the machines and their ethernet
adaptors are up, move on to the next step.

Step 7: Complete Cluster Setup

Step 8: Test Cluster Setup

The following buttons are for managing
your node definitions after the initial install.

Add OSCAR Clients...

Delete OSCAR Clients...

Install/Uninstall OSCAR Packages...

source: **http://gchelpdesk.ualberta.ca/news/20jan05/cbhd_news_20jan05.php**

If any step failed to execute then go to the concerned site for reporting the bugs or see the reply of the previously reported bugs and find out the solutions.

In this step only you can define the number of clients you want to add in cluster.

You have to assign an individual IP addresses to all your clients. Also you have to assign MAC addresses to all the nodes after which network boot is to be done in order to make the clients synchronize with the servers . After that run the step complete cluster setup to make the clustering process over and then check the setup.

Hence clustering is done in a very simple and easy manner.

The main benefit of using this package is that there is no need to configure or install the different file systems or services which otherwise will have to install separately.

All you have to do is to fire a job on the server node and the server itself manages to distribute it to Client nodes, so that the user need not bother about this.

You can add or remove any number of clients at any instance, even after the setup is done.

General Information:

ADDING USERS
When you add users you will need you make sure that they can ssh unchallenged from the server to any node and also the other way round.

To add user the Command is adduser or useradd and username.

eg; useradd john

to change password of the user type 'passwd john' and enter the new password.

You can go to home directory of any user from the server by typing

# su – username

Conclusion :

So, clustering is done in an efficient manner by following the above given steps and instructions .The prime concern for making this documentation is to introduce a general idea of utilizing the available resources by making the users aware of the knowledge and concepts of Linux which otherwise would not be explored due to lack of proper training and guidance .Also,while doing clustering one will be well versed with Linux operating system ,which is considered as a bit complex and much more technical than any other operating system.