

# A Recipe for a diskless MOSIX cluster using Cluster-NFS

*DRAFT*

Gregory R. Warnes  
Fred Hutchinson Cancer Research Center

May 10, 2000

## 1 Server Configuration

### 1.1 Install and configure Debian Linux

I used a Corel Linux CDROM ( <http://www.corel.com> ), and then upgraded to a more recent version of debian using `dselect` , but any Debian installation should work fine ( <http://www.debian.org> )

### 1.2 Download, Compile, Install Cluster-NFS

Source and debian packages are available from <http://queenbee.fhcr.org/Cluster-NFS/>

If you are running a debian potato or more recent debian distribution,

```
cd /tmp
wget http://queenbee.fhcr.org/Cluster-NFS/Cluster-NFS.current.deb
dpkg install Cluster-NFS.current.deb
```

## 1.3 Download, Unpack, Install Kernel and MOSIX

Kernel Configuration, kernel 2.2.15 used as example

- Download and Unpack Kernel

```
cd /usr/src
wget http://www.us.kernel.org/pub/linux/kernel/v2.2/linux-2.2.15.tar.gz
tar -xvzf linux-2.2.15.tar.gz
```

- Download, Unpack, Compile and Install MOSIX for the Server

The MOSIX bundle is available from <http://www.mosix.cs.huji.ac.il>

The MOSIX installation program also builds the kernel sources, and allows you to configure the kernel. The first time through configure the kernel for the server machines. IE, *DO NOT* enable BOOTP or rootNFS.

```
cd /usr/src
wget http://www.mosix.cs.huji.ac.il/ftps/MOSIX-0.97.4.tar.gz
mkdir MOSIX
cd MOSIX
tar -xvzf ../MOSIX-0.97.4.tar.gz
./mosix.install
```

- Reconfigure the kernel for the clients, enabling BOOTP and rootNFS.

```
cd /usr/src/linux
make mrproper
make menuconfig ## or make config or make xconfig
```

To enable BOOTP and rootNFS, set the kernel configuration options:

```
[Code Maturity Options] → [Prompt for development and/or incomplete code/drivers]
[Networking Options] → [IP: kernel level autoconfiguration]
[Networking Options] → [IP: kernel level autoconfiguration] → [BOOTP support]
[Filesystems] → [Network File Systems] → [NFS filesystem support]
[Filesystems] → [Network File Systems] → [NFS filesystem support] → [Root file
system on NFS]
```

- Compile the kernel

```
make dep
make bzImage
```

## 1.4 Make Boot Floppies

Insert a formatted floppy in the drive and use “dd” to copy the kernel image to the floppy:

```
cd /usr/src/linux
dd if=arch/i386/boot/bzImage of=/dev/fd0
```

## 1.5 Add entries for each client server files

- `/etc/mosix.map` ,

On our system, the cluster nodes are assigned ip numbers 10.0.0.1 - 10.0.0.8 so this file only needs one range:

```
1 10.0.0.1 8
```

- `/etc/hosts` ,

On our system, the cluster nodes are named bee1 through bee8, so `/etc/hosts` includes the lines

```
10.0.0.1      bee1      biostat1    b1
10.0.0.2      bee2      biostat2    b2
10.0.0.3      bee3      biostat3    b3
10.0.0.4      bee4      biostat4    b4
10.0.0.5      bee5      biostat5    b5
10.0.0.6      bee6      biostat6    b6
10.0.0.7      bee7      biostat7    b7
10.0.0.8      bee8      biostat8    b8
```

- `/etc/bootptab` ,

The BOOTP daemon uses `/etc/bootptab` to respond to the queries made by the compute nodes using the ethernet MAC address of the node.

Contents our `/etc/bootptab` :

```
.beehive:\
    :bf=null:\
    :dn=beehive.fhcrc.org:\
    :ds=10.0.0.1:\
    :gw=10.0.0.1:\
    :hd=/:\  
    :hn:\
    :nt=10.0.0.1:\
    :rp=/:\  
    :sm=255.255.255.0:\
    :to=auto:

bee2:ht=1:ha=00A0C98A1040:tc=.beehive:
bee3:ht=1:ha=00A0C98CF35A:tc=.beehive:
bee4:ht=1:ha=00A0C945BE60:tc=.beehive:
bee5:ht=1:ha=00a0c945c086:tc=.beehive:
bee6:ht=1:ha=000102241106:tc=.beehive:
bee7:ht=1:ha=000102241108:tc=.beehive:
bee8:ht=1:ha=00010224110a:tc=.beehive:
```

- `/etc/exports` , and

```
/    10.0.0.1(rw,no_root_squash)
/    10.0.0.2(rw,no_root_squash)
/    10.0.0.3(rw,no_root_squash)
/    10.0.0.4(rw,no_root_squash)
/    10.0.0.5(rw,no_root_squash)
/    10.0.0.6(rw,no_root_squash)
/    10.0.0.7(rw,no_root_squash)
/    10.0.0.8(rw,no_root_squash)
```

- /etc/hosts.allow.

ALL: 10.0.0.1

ALL: 10.0.0.2

ALL: 10.0.0.3

ALL: 10.0.0.4

ALL: 10.0.0.5

ALL: 10.0.0.6

ALL: 10.0.0.7

ALL: 10.0.0.8

## 1.6 Create files that are the same for all clients.

Files named `filename$$CLIENT$$` will be served when any client requests the file `filename`

On our system, we have

<code>/etc/exports\$\$CLIENT\$\$</code>	empty file
<code>/etc/fstab\$\$CLIENT\$\$</code>	see below
<code>/etc/inetd.conf\$\$CLIENT\$\$</code>	comment out bootp entry
<code>/etc/mospe\$\$CLIENT\$\$</code>	soft link to a non-existent file
<code>/etc/nsswitch.conf\$\$CLIENT\$\$</code>	remove "dns" from hosts entry
<code>/etc/resolv.conf\$\$CLIENT\$\$</code>	remove name server entries
<code>/etc/syslog.conf\$\$CLIENT\$\$</code>	see below
<code>/etc/init.d/apache\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/apache-ssl\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/apcd\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/atd\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/cron\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/elmd\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/exim\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/gdm\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/genpower\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/gpm\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/lpd\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/mountall.sh\$\$CLIENT\$\$</code>	see below
<code>/etc/init.d/network\$\$CLIENT\$\$</code>	see below
<code>/etc/init.d/proftpd\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/etc/init.d/xfss\$\$CLIENT\$\$</code>	soft link to <code>DoNotExecuteOnClients</code>
<code>/usr/sbin/ssmtp\$\$CLIENT\$\$</code>	see below

Specific files

- `/etc/syslog.conf$$CLIENT$$`

Forward all syslog information to the master node `queenbee` .

```
# forward everything to queenbee
```

```
*.*
```

```
@queenbee
```

- /etc/init.d/mountall.sh\$\$CLIENT\$\$

This file needs to be modified so that it mounts the root nfs filesystem.

```

--- /etc/init.d/mountall.sh      Mon Mar 13 10:07:59 2000
+++ /etc/init.d/mountall.sh$$CLIENT$$  Sat Mar 25 01:40:17 2000
@@ -8,8 +8,8 @@
#
# Mount local file systems in /etc/fstab.
#
-[ "$VERBOSE" != no ] && echo "Mounting local file systems..."
-mount -avt nonfs,noproc,nosmbfs
+[ "$VERBOSE" != no ] && echo "Mounting ALL file systems..."
+mount -avt noproc,nosmbfs

#
# We might have mounted something over /dev, see if /dev/initctl is there.
```

- /etc/init.d/network\$\$CLIENT\$\$

Usually, this file is used to start up the network connections. On the clients the network is already started by the time we get this far, so all that needs to be done is to set up the loopback device:

```

echo "Setting up network "
echo "  current status "
ifconfig
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

- /usr/sbin/ssmtp\$\$CLIENT\$\$

I have not set up the nodes to connect to the outside, so mail creation requests need to be forwarded to the master node ( qb is an alias for queenbee ):

```

#!/bin/bash
# funnel all email to queenbee, which forwards it all on to lynx

ssh qb /usr/sbin/ssmtp $*
```

## 1.7 Create files that are specific to individual clients

Files named `filename$$IP=xxx.xxx.xxx.xxx$$` will be served to the client with ip `xxx.xxx.xxx.xxx` when it requests `filename\`.

On the BioHive cluster, the following per-node files exist:

<code>/dev\$\$IP=10.0.0.2\$\$</code>	may not be necessary
<code>/etc/X11/XF86Config\$\$10.0.0.2\$\$</code>	depends on VGA card used
<code>/etc/adjtime\$\$IP=10.0.0.2\$\$</code>	may not be necessary
<code>/etc/devices\$\$IP=10.0.0.2\$\$</code>	modified on bootup by each machine
<code>/etc/hostname\$\$IP=10.0.0.2\$\$</code>	machine specific
<code>/etc/mtab\$\$IP=10.0.0.2\$\$</code>	created by “mount” on each machine
<code>/etc/rmtab\$\$IP=10.0.0.2\$\$</code>	created by “mount” on each machine
<code>/tmp\$\$IP=10.0.0.2\$\$</code>	needed for temp files, lock files
<code>/var\$\$IP=10.0.0.2\$\$</code>	needed for lock files, spool files

Most of the files can be created by `echo > filename\$\$10.0.0.2\$\$`. The directories should be created by `cp -r \dirname \dirname\$\$IP=10.0.0.2\$\$`

## 1.8 Reboot server to restart all services