# Netexplo Research Group Emulation Cluster Guide

Department of Computer Science
The University of Hong Kong

Yu Wu, Linquan Zhang

May 5, 2011

# Chapter 1

# Overview

## 1.1 Architecture

The architecture of the emulation cluster is given in Fig. 1.1.

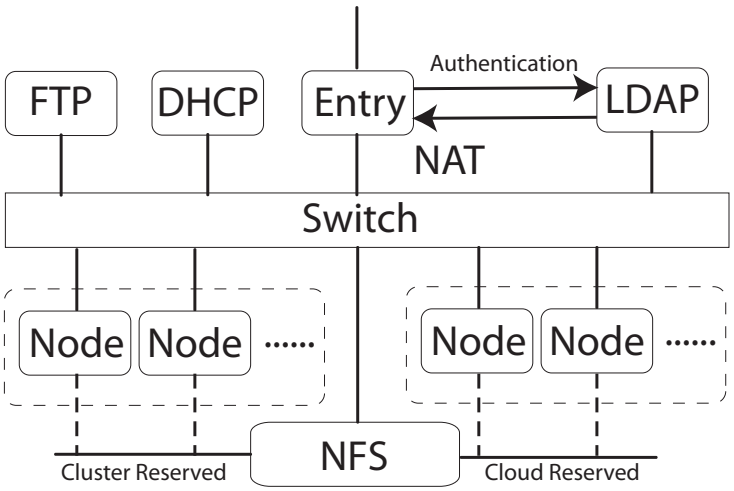


Figure 1.1: Architecture of the emulation cluster.

**Entry node**: the gateway to the cluster, which ideally should contain two NICs — eth0 and eth1. eth0 is configured with a public IP address (**202.45.128.129**), which is the only entry point to the cluster. eth1 is connected to the switch with a private IP address (192.168.1.1). Currently, the entry node we are using has only one NIC, *i.e.*, eth0, and another NIC is virtualized as a workaround. For future upgrade, the entry node is preferably replaced with a more powerful machine, since it may constitute the performance bottleneck when large-scale jobs are executed on the cluster. For the same reason, the entry node should only handle necessary external requests and access control functionalities, *e.g.*, firewall. A firewall with default rules has been set up on the entry node, which will be further customized when need arises. **Note: NEVER run any of your simulation/emulation programs on the entry node.**

**Working nodes**: all configured with private IP addresses (ranging from 192.168.1.3 to 192.168.1.252). The working nodes are divided into two groups:

- Cluster-reserved machines: about 24 machines are allocated for regular emulation usage, with IP addresses ranging from 192.168.1.200 to 192.168.1.223. They are installed with clean Fedora 8.

- Cloud-reserved machines: about 40 machines are allocated to construct a cloud system, with IP addresses ranging from 192.168.1.3 to 192.168.1.41. These machines are installed with Xen-bundled Fedora 8.

Refer to Chapter 4 for hardware details of the nodes.

**NFS server**: A Dell PowerEdge T410 server at 192.168.1.253, where all users' data are stored. When a user logs in to the cluster, NFS directories belonging to the specific authenticated user will be mounted to the machines allocated to the user automatically (a process transparent to the users). The NFS server is configured with RAID-5 for performance optimization and failure tolerance.

## 1.2 Key Services Installed

The important services installed in the emulation cluster include:

1. **FTP** is used to copy disk image for each working node in cases of failure recovery and fast deployment. There are two images: one is for cluster-reserved nodes and the other is for cloud-reserved ones. Compared with other disk ghost method, *e.g.*, "dd", we think this is the most efficient way after our careful assessment. Using FTP, finer-grained backup can also be achieved for individual working node, and this can be implemented in future if needed.

2. **DHCP** is used to manage and assign IP addresses to machines when they boot up. We set up the simple static mac-IP mapping policy to ease the management and monitoring tasks in a centralized fashion. An alternative policy is to set up an IP pool and dynamically assign each newly-booted machine with an IP within the pre-set range.

3. **LDAP** is in charge of user account management and user authentication. When a user logs in to the gateway (entry node), he will automatically log in to all the machines allocated to him, *e.g.*, he can "ssh" to any allocated machine freely.

Currently, all the services are installed on the Dell NFS server (192.168.1.253), except that FTP server for copying the cluster-reserved node image is on Machine 192.168.1.2.

# Chapter 2

# Technical Details

## 2.1 Xen-based Virtualization

We include important Xen virtual machine installation techniques in this section.

### 2.1.1 Hardware Compatibility Test

To support virtualization, the CPU in a machine must, at a minimum, supports Physical Address Extension (PAE). To verify if a CPU supports PAE, run

**grep pae /proc/cpuinfo**

To implement full virtualization, the CPU must include intel-VT (vmx) or AMD-V (svm) support, to be verified as follow

**grep vmx /proc/cpuinfo**

**grep svm /proc/cpuinfo**

Since our machines do not meet this requirement, we implement para-virtualization.

### 2.1.2 OS Modification

Among the free operating systems, Fedora and CentOS both support Xen well (but not for higher versions of Xen). However, Xen can not be launched directly on a standard kernel, since it does not support well enough various hardware configurations, especially new hardwares. We have created a modified kernel based on Fedora 8. As stated earlier, the kernel image for cloud-reversed nodes is stored on the FYP server, and bit-by-bit copied to install on the cloud-reserved working nodes. One thing to note is that, after kernel installation via the network, the IP address and host name should be modified from the default ones in the image, to avoid naming conflict.

We use CentOS with a modified kernel as the guest OS running on top of Xen.

### 2.1.3 Xen Bridging

It will be helpful to gain a deeper understanding on Xen's underlying mechanism of virtualization.

#### Xen virtualization mechanism

When xend starts up, it runs the network-bridge script, which:

1. Creates a new bridge named xenbr0
2. "Real" ethernet interface eth0 is brought down
3. The IP and MAC addresses of eth0 are copied to virtual network interface veth0
4. Real interface eth0 is renamed to peth0
5. Virtual interface veth0 is renamed eth0
6. peth0 and vif0.0 are attached to bridge xenbr0.
7. The bridge, peth0, eth0 and vif0.0 are brought up.

When a domU starts up, xend (running in dom0) runs the vif-bridge script, which:

1. Attaches vif<id#>.0 to xenbr0
2. vif<id#>.0 is brought up.

#### Xen-bridging configuration

Xen sets default bridge as "eth0", which we should change to bridge to other devices when needed in future by modifying the file **/etc/xen/xend-config.sxp**: uncomment "**network script 'network-bridge netdev=eth1'**" and restart xend. After that, verification can be done by "ifconfig peth1"

#### VM's configuration

In our case, we only deal with the "pygrub.cfg" file, which includes

*v*if = ['bridge=eth1, mac = xx:xx:xx:xx:xx:xx']

*d*isk = ['file /root/centos...]

Besides, before launching VMs, make sure SELinux is disabled. To temporarily disable SELinux, run

**setenforce 0**

### 2.1.4 Some Tips

**How to install Xen on Fedora8**

1. yum groupinstall Virtualization
2. Edit **/boot/grub/grub.conf** and choose xen as the default boot option
3. Reboot

**How to install paravirtualized DomU via text mode**

1. mount -t iso 9660 fedora-XXX.iso /var/www/html/fedora -o loop, ro
2. Disable SELinux temporarily by **setenforce 0**
3. Start up Apache by **service httpd start**
4. Disable firewall by **service iptables stop**
5. Check the IP address of virbr0, which by default is 192.168.122.1
6. Install the paravirtualized domU by **virt-install -n fedora -w network:default -p -f /var/lib/xen/images/fedora.img –nographics -r 512 -s 16 -l http://192.168.122.1/fedora/**
7. Enable SELinux by **setenforce 1**
8. Enable firewall by **service iptables start**
9. Disable Apache by **service httpd stop**

For GUI installation, run: **virt-manager**.

**How to start the Xserver after installing Xen**

A problem was resolved with "xorg.conf". The original file of "xorg.conf" is:

```
Section ‘‘Device’’
    Identifier  ‘‘Videocard0’’
    Driver      ‘‘Intel’’
EndSection
Section ‘‘Screen’’
    Identifier ‘‘Screen0’’
    Device ‘‘Videocard0’’
    DefaultDepth 24
    subsubsection ‘‘Display’’
        Viewport 0 0
        Depth 24
    EndSection
EndSection
```

which should be modified to

```
Section ‘‘Device’’
    Identifier  ‘‘Generic Video Card’’
    Driver      ‘‘vesa’’
EndSection
Section ‘‘Screen’’
    Identifier ‘‘Screen0’’
    Device ‘‘Generic Video Card’’
    DefaultDepth 24
    subsubsection ‘‘Display’’
        Viewport 0 0
        Depth 24
    EndSection
EndSection
```

**About CentOS**

1. When trying yum, remember add "**http://**" when setting up a proxy.
2. When installing an editor like vi or vim, use **yum -y install vi***.

**How to access the Internet from a VM**

1. Set up the IP address and the netmask by running:
   **ifconfig eth0 147.8.177.* netmask 255.255.252.0**
2. Set up the DNS server by running:
   **echo nameserver 147.8.176.15 > /etc/resolv.conf**
3. Set up the proxy by running:
   **export http_proxy=http://proxy.cs.hku.hk:8282**

**After cloning the image**

After copying the OS image from the FTP server, the cloned machine will show 2 NICs. What we need to do first is to remove the old one which contains the NIC info of the source machine from which the image is made (If needed, remove all the registered NICs). The file is located at **/etc/udev/rules.d/70-net_persistent**.

## 2.2 NFS Service

NFS server is deployed on the Dell file server (192.168.1.253) with OpenSUSE 11. Data files of all users are stored on it and can only be modified by their owners. For users' convenience and security, NFS directories belonging to a specific user will be mounted automatically to every single working node allocated to the user. Although cache is enabled on the local machines, any updates to the data files will be written back to the NFS

server. Users' authentication configuration files are also stored on NFS server, which is used for authentication together with the LDAP server.

To enable the NFS server:

1. Edit /etc/exports, adding /export/home (rw)
2. chkconfig –level 35 rpcbind on; chkconfig –level 35 nfslock on; chkconfig –level 35 nfs on

To enable a NFS client:

1. Create the directory /export/home (mkdir -p /export/home)
2. Add NFS server (Edit file /etc/fstab, add one line: 192.168.1.253:/home /export/home nfs defaults 1 1)
3. Disable selinux by modifying /etc/selinux
4. Configure the startup service (chkconfig –level 35 nscd on; chkconfig –level 35 rpcbind; chkconfig –level 35 netfs on)

## 2.3 User management based on LDAP

We manage user accounts and implement authentication with *openLDAP*.

### 2.3.1 Configuring LDAP Server

To enable the LDAP server on OpenSUSE 11:

1. disable TLS.
2. Base DN is set as "dc=intgroup, dc=net".
3. Administrator DN is set as "cn=admin".
4. Database Directory is by default "/var/lib/ldap", whose type is "hdb".

When adding or deleting a user:

1. Create the home directory for the user, "mkdir -p /export/home/xxx"
2. Enter the user account panel, filter "LDAP user"
3. Home directory is set to the newly created "/export/home/xxx" with permission "755"

### 2.3.2 Configuring a LDAP Client

1. Configure LDAP client via authorize-gtk, or authorize-cui
2. Choose the option "Use LDAP"
3. Specify the LDAP server, "192.168.1.253"
4. Fill in the LDAP base DN "dc=intgroup,dc=net"
5. Disable TLS/SSL

## 2.4 Installing OS via PXE

Since not all the machines support booting from USB devices, booting via the network is necessary for OS installation from time to time, for which PXE (Preboot eXecution Environment) service is a nice choice.

### 2.4.1 Configuring PXE Server

To enable PXE server, There are four steps.

1. modify dhcpd.conf, add the following lines:

   subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.55 192.168.1.199;
   option subnet-mask 255.255.255.0;
   option domain-name "intgroup.net";
   **next-server 192.168.1.2;**
   **filename "/pxelinux.0";**
   }

2. configure TFTP server, edit /etc/xinetd.d/tftp as follows:

   service tftp{
   socket_type = dgram
   protocol = udp
   wait = yes
   user = root
   server = **/usr/sbin/in.tftpd**
   server_args = **-s /tftpboot**
   disable = **no**
   per_source = 11
   cps = 1002
   flags = IPv4
   }

3. Extract the PXE Boot loader archive to folder /tftpboot

4. Setup vsFTP server, put Fedora8.iso to /var/ftp/pub/, so that other machines can access the install files via ftp

### 2.4.2 Configuring a PXE Client

1. Restart the PC;
2. When the Dell's logo shows up, press F12;
3. Choose booting up via NIC;
4. Then the installer will setup Fedora 8 OS automatically.

# Chapter 3

# User Guide

This chapter presents the guidance to use the cluster-reserved nodes for regular simulation/emulation purposes.

## 3.1 Reservation System

Before using the cluster, a user is required to book the nodes needed and periods for dedicated usage of those nodes, via the reservation system at *http://202.45.128.129/calendar*. A snapshot of the reservation system is shown in Fig. 3.1.

Steps for making a reservation:

- Log in the reservation system. The user name and password are the same as your account in the cluster, which will be introduced in Sec. 3.2.
- Add a reservation entry by clicking the top right corner (±) of the time slot(s), in which you wish to use the cluster (Fig. 3.2 ). **You are strongly recommended to check whether your reservation conflicts with those of others.**
- You may edit your reservation entry following what is shown in Fig. 3.3.

There are two ways to access the reservation system from off-campus networks:

1. Start a remote desktop connection to your desktop in the office, then you may use the reservation system via remote desktop.
2. Build a ssh tunnel as follows:
   a) Open Terminal
   b) ssh -L 10002:ywu-gw:80 xxx@gatekeeper.cs.hku.hk (replace xxx with your CS user name)
   c) Open browser
   d) In the address bar, type: http://localhost:10002/calendar/
   (you can change 10002 in b) and d) to other available port number on your computer)

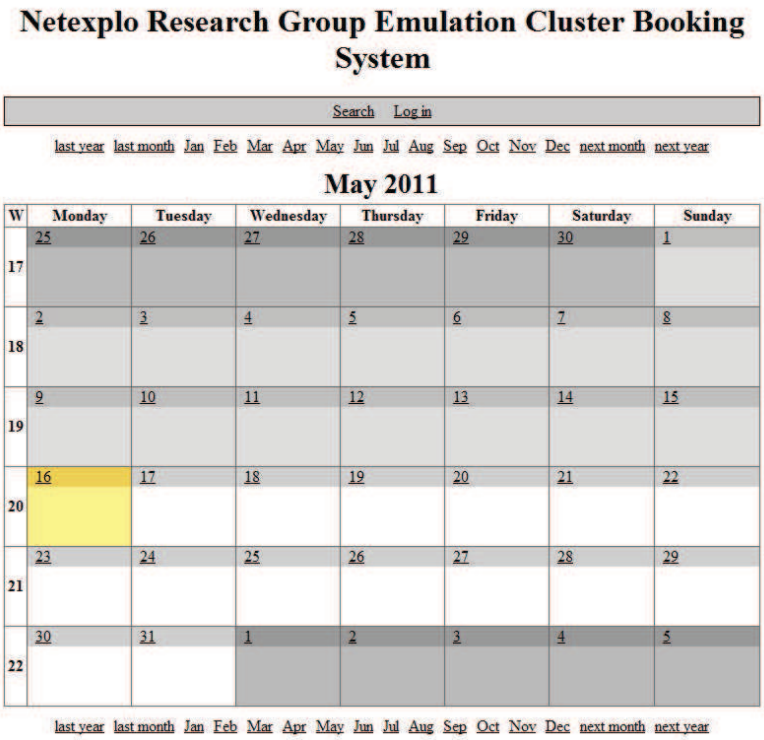


Figure 3.1: Cluster reservation system.

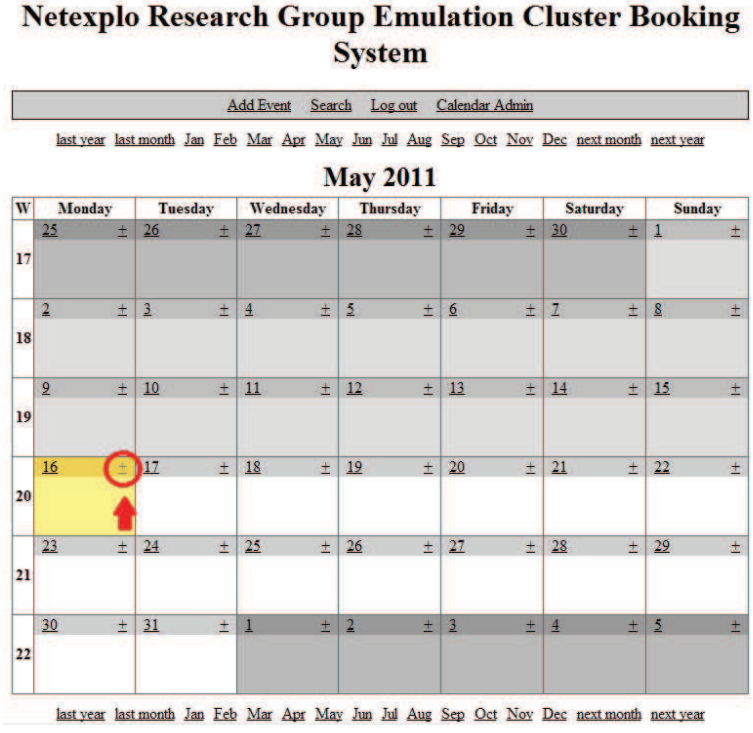


Figure 3.2: Cluster reservation system: making a reservation.

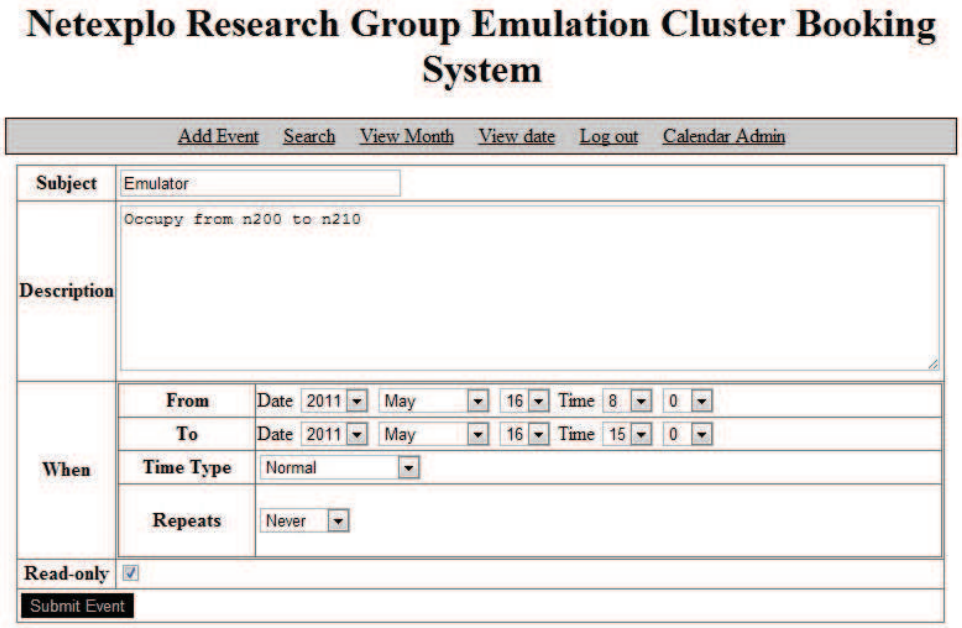


Figure 3.3: Cluster reservation system: editing a reservation.

## 3.2 How to Connect to the Cluster

Users can access the cluster via SSH to the entry node at IP address *202.45.128.129*. If you are not in HKU network, you should log in *gatekeeper.cs.hku.hk* using your CS account first.

Your username is the same as your CS account username, and the initial password is set to the same as the username. For example, Jian Zhao's username and initial password are both *jzhao*. You are strongly recommended to change your password as soon as possible, which you can do in the following way:

**[jzhao@n0]# passwd**

## 3.3 How to Deploy your Program in the Cluster

There are currently 24 machines available in the cluster-reserved node group. Their hostnames are *n200, n201, ... n223*, respectively. If you wish to use node *n200*, you should first log in to the entry node via SSH, then type "ssh n200", and then you are logged in to node *n200*.

**Please note: NEVER run any of your simulation/emulation programs on the entry node, which is only meant as an entrance for everyone to access the cluster and is installed with important services to be used by everyone. Running any program on the entry node will lead to significant performance downgrade of the cluster. Log in your reserved working nodes and run your program there instead.**

If you wish to use multiple machines to run the same program, you can do it in the following fashion:

**[jzhao@n0]# for((i=200;i<=223;i=i+1)) do ssh n$i program; done**

## 3.4 Service and Software Available on Each Node

- Services: SSH, SVN, HTTP

- Software packages: gcc v4.1.2, Java v1.7.0, Python v2.5.1, Perl v5.8.8, Vim v7.1.135

Note: other needed software packages can be installed upon request.

## 3.5 How to Setup a SVN Server

If you wish to set up a SVN server on a machine allocated for your usage (*e.g.*, to maintain the simulation/emulation code you are developing), follow these steps:

1. Suppose that Jian wishes to create a repository for files in the /src directory on the machine. First create the directory "~/src" by:

   **[jzhao@n0]# mkdir /src**

2. Use the svnadmin command to create the SVN repository within this directory:

   **[jzhao@n0]# svnadmin create /src**

3. Edit the file svnserve.conf in ~/src/conf to:

   **anon-access = none**
   **auth-access = write**
   **password-db = passwd**

4. Add user & password into the file passwd in ~/src/conf:

   **[users]**
   **jzhao = password**

5. Start SVN server by:

   **[jzhao@n0]# svnserver -d**

## 3.6 Recommended Websites on Linux

1. Basic knowledge of Linux: `http://linux.vbird.org`
2. Basic knowledge of Vim: `http://vimdoc.sourceforge.net/`
3. Advanced Shell Programming: `http://www.gnu.org/software/bash/manual/bashref.html`

# Chapter 4

# Hardware Details

## 4.1 Switch

Currently the cluster employs three switches to interconnect the machines:

- one IBM Fast Ethernet Desktop Switch 8275 Model 324 (as shown in Fig. 4.1), which offers 24 ports helping drive 10/100 switching. Port trunking capabilities provide aggregate bandwidth (up to 800Mbps) between the switches.
- one Intel Express 510T switch, with 24 Ethernet 10Base-T/100Base-TX ports, supporting 8K MAC addresses, and 100Mbps data transfer rate.
- one linksys SRW2024 switch, with 24 Ethernet 10Base-T/100Base-TX/1000Base-T ports, supporting 10/100/1000Mbps data transfer rate and 48Gbps backplane bandwidth.


Figure 4.1: IBM Fast Ethernet Desktop Switch.

## 4.2 Cluster-Reserved Nodes

Table 4.1: Configurations of the Cluster-Reserved Nodes

| **Hostname** | **Configuration** |
|---|---|
| n200 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n201 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n202 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n203 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.40GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n204 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n205 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n206 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n207 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n208 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n209 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |

| Hostname | Configuration |
|---|---|
| n210 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n211 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n212 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n213 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n214 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n215 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n216 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n217 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n218 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n219 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n220 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |

| **Hostname** | **Configuration** |
|---|---|
| n221 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n222 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |
| n223 | processor : 0<br>model name : Intel(R) Pentium(R) 4 CPU 2.26GHz<br>cache size : 512 KB<br>MemTotal: 1 GB |