

# A simple installation and administration tool for the large-scaled PC cluster system: DCAST

Tomoyuki HIROYASU, Mitsunori MIKI, Kenzo KODAMA,  
Junichi UEKAWA, and Jack DONGARRA

ISDL  同志社

Doshisha University

Intelligent Systems Design Laboratory



**Innovative Computing Laboratory**

COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF TENNESSEE

# PC Cluster Features

## **Good cost-performance ratio**

It is possible to create a PC cluster cheaply by combining mass-produced parts. Also, by improving computational power of these parts, the cost/performance ratio is good.

## **Can construct with arbitrary size**

It is possible to construct any size of PC cluster by changing the number of nodes that construct the PC cluster.

## **Possible to construct with arbitrary combination**

It is possible to combine different CPUs in a PC cluster by placing different CPUs on different nodes. Also, it is possible to construct PC clusters with some nodes not having hard drives.

## **Possible to introduce newer technologies**

It is possible to introduce new technologies that are brought into PCs, such as Gigabit Ethernet and Hyper-threading CPU. Also, it is possible to introduce latest software technology such as communication middleware and compilers.

# Problems with PC clusters

## **Requires knowledge in construction and maintenance**

Involved knowledge of operating systems and networking is required for PC cluster construction and maintenance, and it is a steep learning curve for novices.

## **Installation effort**

PC cluster installation requires operating system installation on each node. Then, extra software and network configuration for operation as PC cluster is required. As PC cluster scales get larger, the configuration process alone becomes a large burden.

## **Maintenance cost**

Each nodes that constructs a PC cluster operates as one computer, and requires same amount of care as a standalone PC, such as software installation, hardware checking, software upgrading, and handling of hardware failures. As PC clusters get larger, the difficulty of finding the problem nodes and keeping consistency between nodes increase.

# Our Facilities



- » 64 nodes
- » P3 1GHz Dual CPU
- » Myrinet 2000



- » IBM IntelliStation 6850-60J
- » 64 nodes
- » Dual CPU of Xeon 2.4GHz
- » Myrinet 2000
- » 320GFlops



- » 256 nodes
- » P3 800MHz Dual CPU

# Our situations...

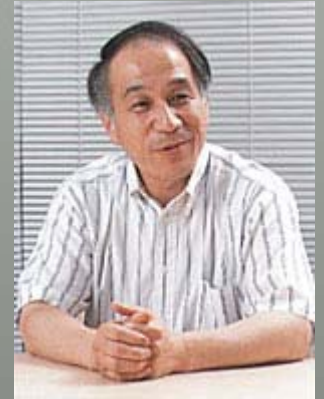
- » We have bought machines ....
- » But, we don't have enough fund for good administrators.
  
- » In Japan,
- » Not only us, but most labs in universities does not have special administrators.
- » Even in companies, there are no administrators.

# Background

- » Generally, writing graduation thesis is required for both bachelor and master course students in Japan.
- » Japanese students have to belong to the lab.
- » There are a lot of students and a few faculties in a lab.
- » Especially in private universities in Japan.

# Background

- » Doshisha University
- » Faculty of Engineering
- » Department of Knowledge Engineering and Computer Sciences
  
- » Faculty Members 2
  - » Professor Mitsunori Miki
  - » Associate Professor Tomo Hiroyasu
- » Doctor Course Students 1
- » Master Course Students 30
- » Bachelor Course Students 20



# Background

- » Therefore, we have to ask our students to administrate our computes, systems, and PC clusters.
- » Students do not have enough knowledge and experience.
  - » This means that we have to educate students as administrators.
- » Students will graduate in two or three years.
  - » Before getting enough knowledge, they are gone.
  - » Students are getting their knowledge gradually
  - » Very efficient system is needed.

# When I was a Graduate Student

---

- » I was also an administrator for the lab.
- » Because I lacked the skills and knowledge, I could not admin the systems very well.
- » I was so scared to stop the system.
- » I stores the batch files.
- » When we wanted to change something for OS or applications, we have just waited that vendors changed.

# After Linux/x86 becomes popular ...

---

- » System is not expensive.
  - » If there are problems after the version up, I can reinstall.
  - » Network Install.
  - » Since there are lot of developers, when I want to install some applications, some one has already made it as a package.
- 
- » Many users
  - » Strong Package system
  - » Many packages

# Debian GNU/Linux

- » Strong package system.
  - » Dependency check
- » Very huge varieties of packages.
- » Frequent update is occurred especially for the packages that have the security problem.
- » Version up is very easy.
- » The system just after the installation does not have many applications.
- » Most options are closed for the Net.

# apt-get

- » Apt-cache search **keyword**
- » Apt-get install **package name**
- » /etc/apt/source.list

```
deb ftp://security.debian.org/debian-security/ stable/updates main contrib non-free  
deb ftp://ftp.jp.debian.org/debian/ stable main contrib non-free
```

# When we use Debian ...

---

- » We can follow the newest system very easily.
- » We can keep the basic securities.
- » We can install the applications as soon as we want to do.
- » We need not to wait the new version for the whole release.

# Installer/ Administration tool for PC Clusters

## NPACI Rocks Cluster Distribution

<http://rocks.npaci.edu/>

San Diego Supercomputer Center, UCSD, Millennium Group at UC Berkeley , Linux Competency Centre in SCS Enterprise Systems Pte Ltd in Singapore , The Open Scalable Cluster Environment in Thailand

## Oscar

<http://oscar.sourceforge.net/software.php>

Bald Guy Software , Dell , IBM , Indiana University , Intel , MSC.Software , NCSA , Oak Ridge National Laboratory , Sherbrooke University

## XCAT

<http://www.alphaworks.ibm.com/tech/xCAT/>

IBM

## SCORE

<http://pdswww.rwcp.or.jp/score/dist/score/html/ja/index.html>

RWCP, PC Cluster Consortium

# The problems of novice administrators

- » Cannot handle the details of many interactive operations required for setups while installing the operating system
- » Cannot keep consistency of software between individual nodes
- » Does not know which software to upgrade
- » Wishes to improve security to a certain level, but does not know how
- » Cannot provide information matching different computer architectures and characteristics

# Requirements for our installer

- » A specification that allows a choice from several architectures may confuse the system administrator.
  - » Homogeneous architectures (only x86)
- » The administrator will not know which option to choose, even when there are interactive questions requiring a choice to be made.
  - » No interactions during the operations
- » Contrary to the technical level of the administrator, the desire to include new technology is high.
  - » Use Debian package
- » Want to consider security if possible.
  - » Use Debian Package
- » It is assumed that our administrators has skills of installing a single PC and rebuilding kernels.

# DCAST and Design Goal (1)

## Doshisha (distributed) Cluster Auto Setup Tool: (DCAST)

### **1. No interactivity at installation time**

Operating system installation requires a lot of interaction. A novice administrator cannot handle such interaction. Also, for constructing a large scale PC cluster, repeated interactive operations become require a large amount of human effort to process. For this reason, DCAST lacks interactive operation completely, and automates installation, to simplify the process of PC cluster construction and to reduce workload of the administrator.

### **2. For upgrading, reconstruct the whole PC cluster**

It is possible to introduce new technology to PC cluster. To do so, upgrading software is required. However, upgrading often makes keeping consistency between nodes of the PC cluster. To solve such problem, DCAST will reconstruct the whole PC cluster on upgrading.

# Design Goal (2)

## 3. Use Debian GNU/Linux

Debian GNU/Linux has an advanced package controlling mechanism, and it is possible to easily upgrade installed software and apply security updates on individual nodes. Also, it tracks conflicts and dependencies between packages it is easier to maintain consistency of software. This kind of package maintenance tool is specific to Debian GNU/Linux, and compared to other distribution, security updates and software upgrades can be performed with ease. With DCAST, Debian GNU/Linux is used.

## 4. Do not assume heterogeneous environment

DCAST assumes novices such as students as a user, and does not consider heterogeneous cluster environment with computers of different architectures. All nodes are assumed to be of x86 architecture.

# Design Goal (3)

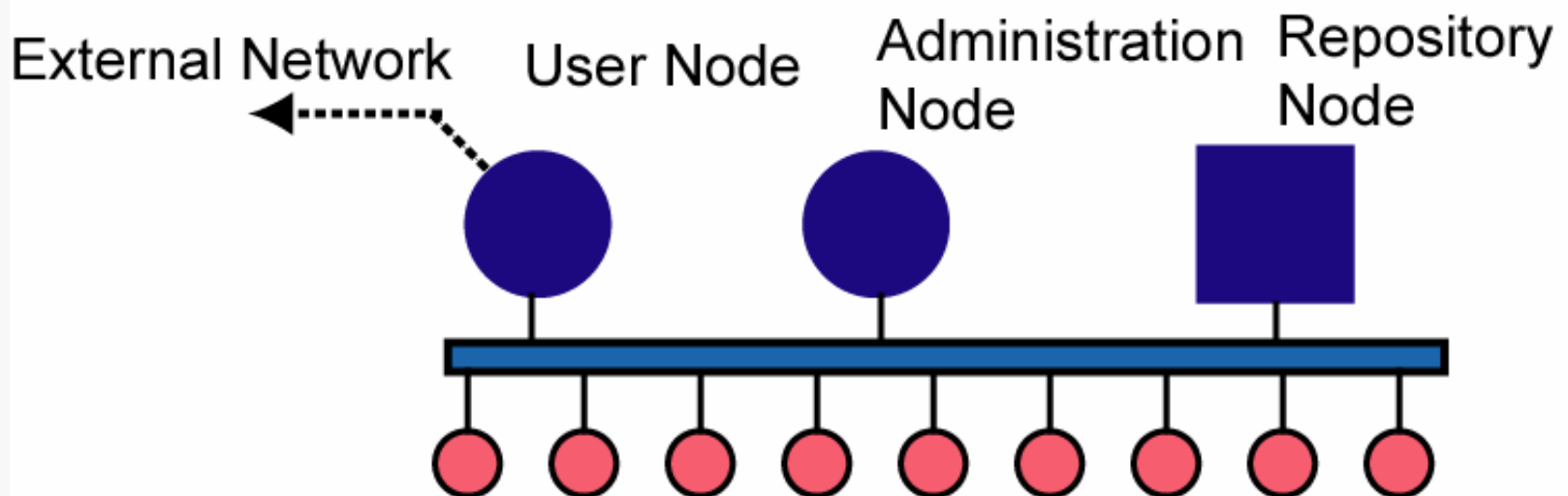
## 5. Allows both diskfull and diskless

Hard drives are moving parts, and they often break. Having many hard drives is not desirable in the view of PC cluster maintenance. To lighten the maintenance cost, there is a form of PC cluster where slave nodes do not have a local hard drive. Such node is called diskless node, and a PC cluster constructed by a master node with disk and diskless nodes is called diskless cluster. The nodes with hard drive are called diskfull nodes, and PC cluster which consists of diskfull nodes as diskfull cluster. DCAST is able to construct either type of PC cluster.

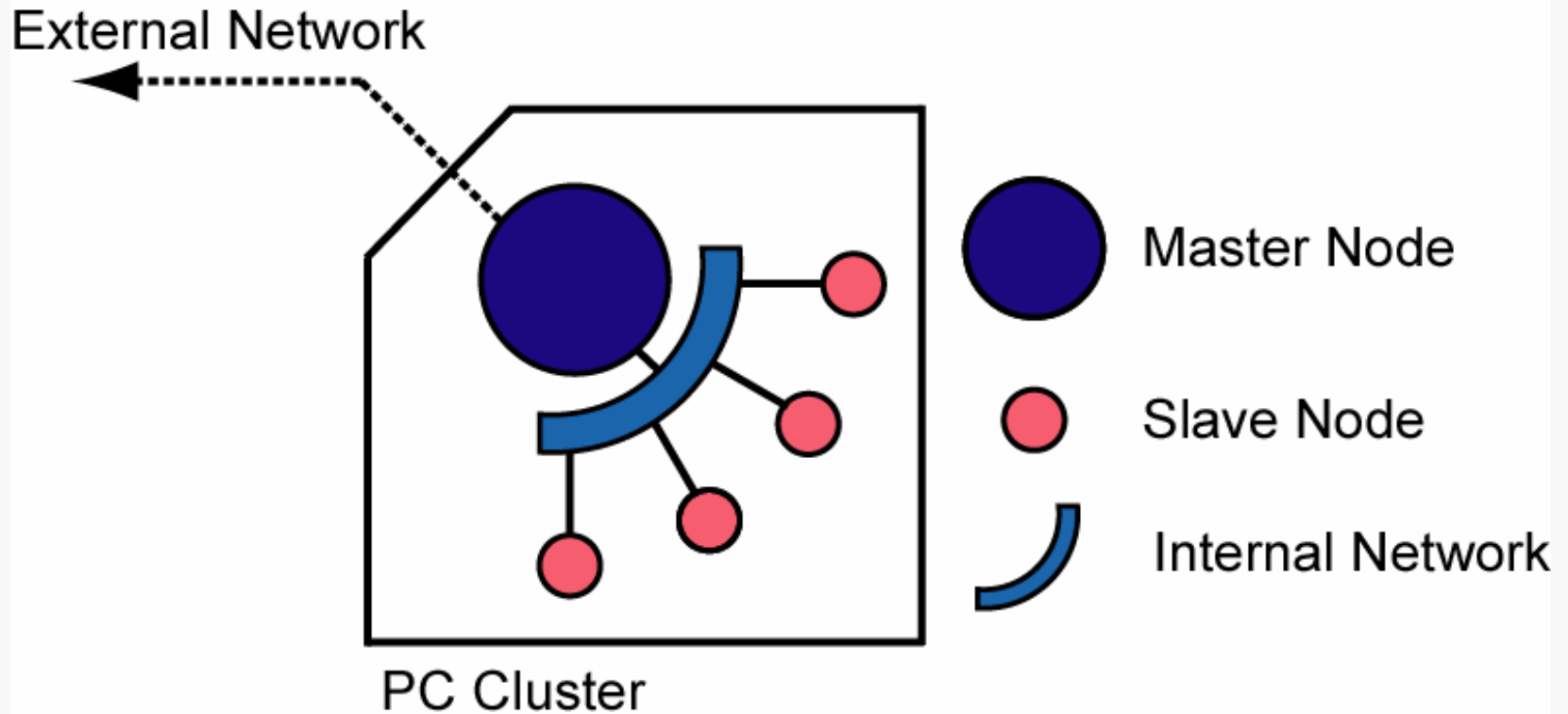
## 6. Integration with existing software

DCAST is constructed using several existing software. By using several software, DCAST software operation is divided, and results in a structure where applying improvements and bug fixes are easier.

# PC Cluster Configuration



# PC Cluster Configuration (simplified)



# Procedures of DCAST

- » Install the master node.
- » Prepare information for slave nodes
- » Prepare data for slave nodes
- » Make boot FDs for slave nodes
- » Boot each slave node
- » Update

# Install the master node

- » **Install the master node.**
- » Prepare information for slave nodes
- » Prepare data for slave nodes
- » Make boot FDs for slave nodes
- » Boot each slave node
- » Update

This procedure is the same as setting up a Linux box .

Bootp option is needed for Kernel.

bootp nfs-kernel-server tftp tftpd  
rsh-server rsh-client cluser-  
update are needed.

# Prepare information for slave nodes

- » Install the master node.
- » **Prepare information for slave nodes**
- » Prepare data for slave nodes
- » Make boot FDs for slave nodes
- » Boot each slave node
- » Update

The mac address of each node is necessary.

```
tcpdump -e | getmac
```

slave.lst is prepared.

# Slave.lst

```
#Enter PARTITION size.  
FPRT /dev/hda1 128 boot *  
SPRT /dev/hda2 512 swap  
TPRT /dev/hda3 - /  
4PRT none - none
```

```
NISDOMAIN nis.org  
LOCALETHCARD eth0  
# NETWORK NETMASK BROADCAST  
NET 192.168.1.1 255.255.255.0 192.168.1.255  
NFSMASTER host_master 192.168.1.2  
GATEWAY 192.168.1.254
```

```
#DCASTMASTER Master's name Master's IP  
DCASTMASTER host01 192.168.1.11
```

```
#slave's name slave's IP slave's MACaddress  
#Autogenerated by update-cluster  
host02 192.168.1.12 009027D0A80B  
host03 192.168.1.13 004005A06C67  
host04 192.168.1.14 004005A886A5  
host05 192.168.1.15 004005A06427  
host06 192.168.1.16 004005A40DEE  
host07 192.168.1.17 004005A40DEF  
#End update-cluster
```

# Make boot FDs for slave nodes

- » Install the master node.
- » Prepare information for slave nodes
- » Prepare data for slave nodes
- » **Make boot FDs for slave nodes**
- » Boot each slave node
- » Update

DCAST uses a floppy disk with grub image for booting up the slave nodes.

```
dcast-gruboppy (network type) (diskfull or diskless)
```

```
dcast-gruboppy eepr diskfull
```

# Boot each slave node

- » Install the master node.
- » Prepare information for slave nodes
- » Prepare data for slave nodes
- » Make boot FDs for slave nodes
- » **Boot each slave node**
- » Update

DCAST operates as a whole using other software. The software used is as follows:

- bootp
- tftpd
- NFS
- grub
- update-cluster

# Used Software (1)

## **bootp**

bootp is a protocol used by nodes of a PC cluster to boot up from other nodes on the network. Using this protocol, even when hard drive is empty, it is possible to start up a kernel from the network, and obtain an IP address to start up a node. DCAST uses bootpd software to provide this service.

## **tftp**

tftp is a trivial file transfer protocol, and the protocol used to transfer boot images by the request of bootp. This is required for loading the kernel from the network, to boot individual nodes of the PC cluster. Using this protocol, it is possible to boot the kernel on nodes which do not have a kernel image on its own. DCAST uses tftpd software to provide this service.

## **NFS**

Network File System is a protocol to provide server file system to a client through a network, developed by Sun Microsystems. NFS is a de-facto standard protocol which is available on most UNIX-compatible systems. Using this protocol, root file system can be mounted even on machines without a hard drive, and allows construction of diskless PC cluster. DCAST uses nfs-kernel-server software to provide this service.

# Used Software (2)

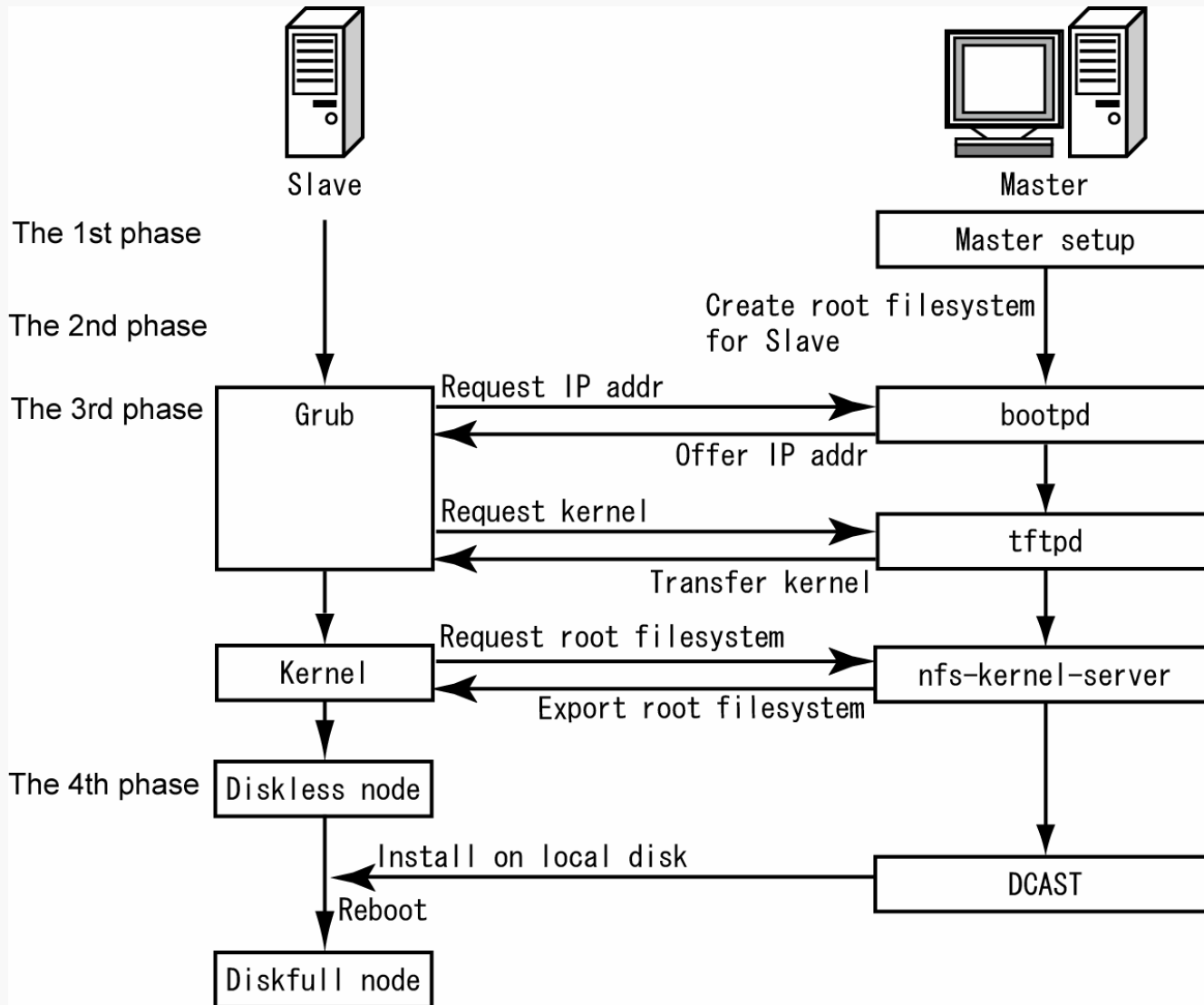
## **grub**

grub is a bootloader for loading the operating system, and can boot up operating system specified in the configuration file. As a functionality of grub, it can analyze the system, so the configuration file can be placed within the system, and it is easy to change configuration. Also, it has functionality as a bootp and tftp client, and it can network boot from the server. DCAST uses a floppy disk with grub image for booting up the slave nodes.

## **update-cluster**

update-cluster is a tool to maintain node information in a uniform manner using XML. In DCAST, by using update-cluster, DCAST configuration file and software configuration file are generated automatically from XML. For example, the configuration file for the popular parallel programming library MPICH can be generated. Also, update-cluster is effective for recreating configuration files when structure of PC cluster has changed.

# Installation procedures



# Update

- » Install the master node.
- » Prepare information for slave nodes
- » Prepare data for slave nodes
- » Make boot FDs for slave nodes
- » Boot each slave node
- » **Update**

We are assuming the following three situations;

- adding new nodes
- Maintenance for starting up the slave nodes
- upgrading software or installing new software

# Adding new nodes

---

To add new nodes to a PC cluster,

the host name,  
IP address,  
and the MAC address

are added to the **slave.lst**.

This operation allows adding of the new node.

# Maintenance for starting up the slave nodes

- » For diskless node
  - » menu.diskless
- » For diskfull node
  - » menu.diskfull
  - » menu.local

When starting with menu.diskfull, the operating system will boot from the master node, and OS installation will start. For PC cluster upgrading, this configuration is used.

When starting with menu.local, the node will start up with the kernel available on the local hard drive, with no interaction with DCAST master node. For operation as diskfull node, start with menu.local.

# The steps for reinstallation

## Shutting down slave nodes

All slave nodes are shut down, then the respective root file system images on the master node are removed. DCAST provides the command `dcast-remove` command for facilitating the shutdown and removing of file system images.

## Upgrading master node

Master node is then upgraded. Software upgrading is done using Debian GNU/Linux package management system. For software that require different settings for master node and slave nodes, it is necessary to use module scripts, which is described later.

## Invoke dcast-setup

Invoke `dcast-setup` and create the root file system for slave nodes. After creating the root file system, boot the slave nodes. The upgrade process is then completed.

## Upgrading kernel of slave nodes

There are cases when kernel needs to be upgraded for handling new hardware. To upgrade the kernel of slave nodes, a new kernel is placed on the directory on the master node that `tftpd` uses. Then, rebooting the diskless node, the new kernel will be loaded. For diskfull nodes, the grub configuration needs to be modified to the one for booting from the master node (`menu.diskfull`), to load the new kernel from the master node. Using this procedure, slave node kernel upgrade can be done.

# Additional module scripts

---

Installation process for software requires an effort that is proportional to the number of nodes of a PC cluster.

Therefore, it is effective to have a system to automatically install and configure applications on all the nodes of a PC cluster when constructing a PC cluster.

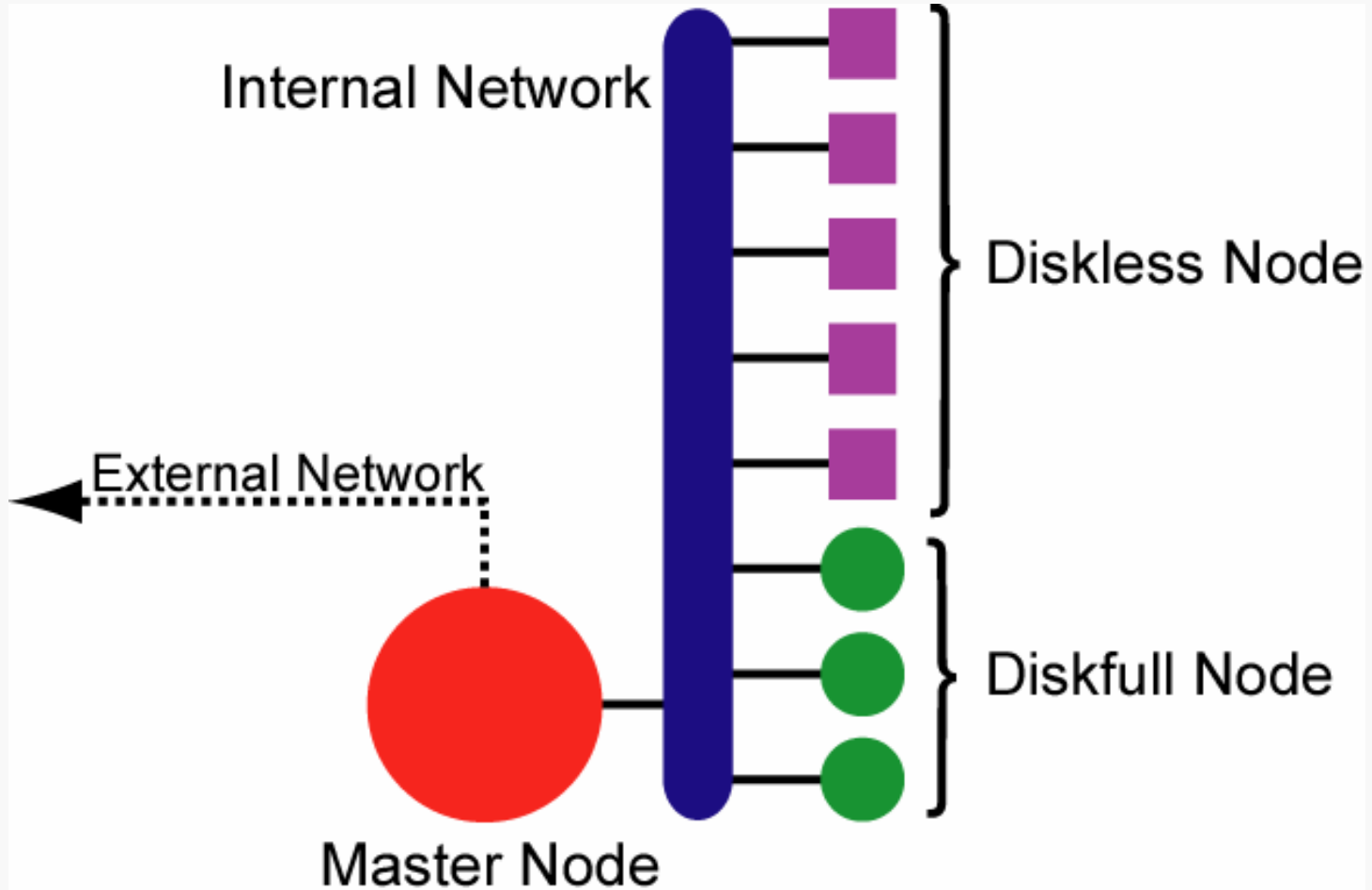
DCAST has a system where different software can be installed when PC cluster is constructed.

# Myrinet Module

To use Myrinet, the following operations are required.

- » Install Myrinet-GM
- » Obtain the MAC address and hostname of the node that is connected via Myrinet, and execute mapper program to initialize the route. A script that does the above, `myrinet-gm.masterconfig` was created.
- » Next, IP address that is used by Myrinet is configured on each slave nodes.
- » A script that does it on each slave node `myrinet-gm.slaveconfig` was created.
- » The created scripts are placed under `/usr/lib/dcast`.
- » By using module scripts, it is possible to use software that requires different settings on each node.

# PC cluster that has a mix of diskless node and diskfull node



# Procedures (1)

---

Step 1. Install Debian GNU/Linux on the master node.

Step 2. Install DCAST.

Step 3. Create grub floppy disk, execute dcast-grubfloppy on the master node, and create grub floppy.

# Procedures (2)

Step 4. Create slave.lst, create slave.lst.

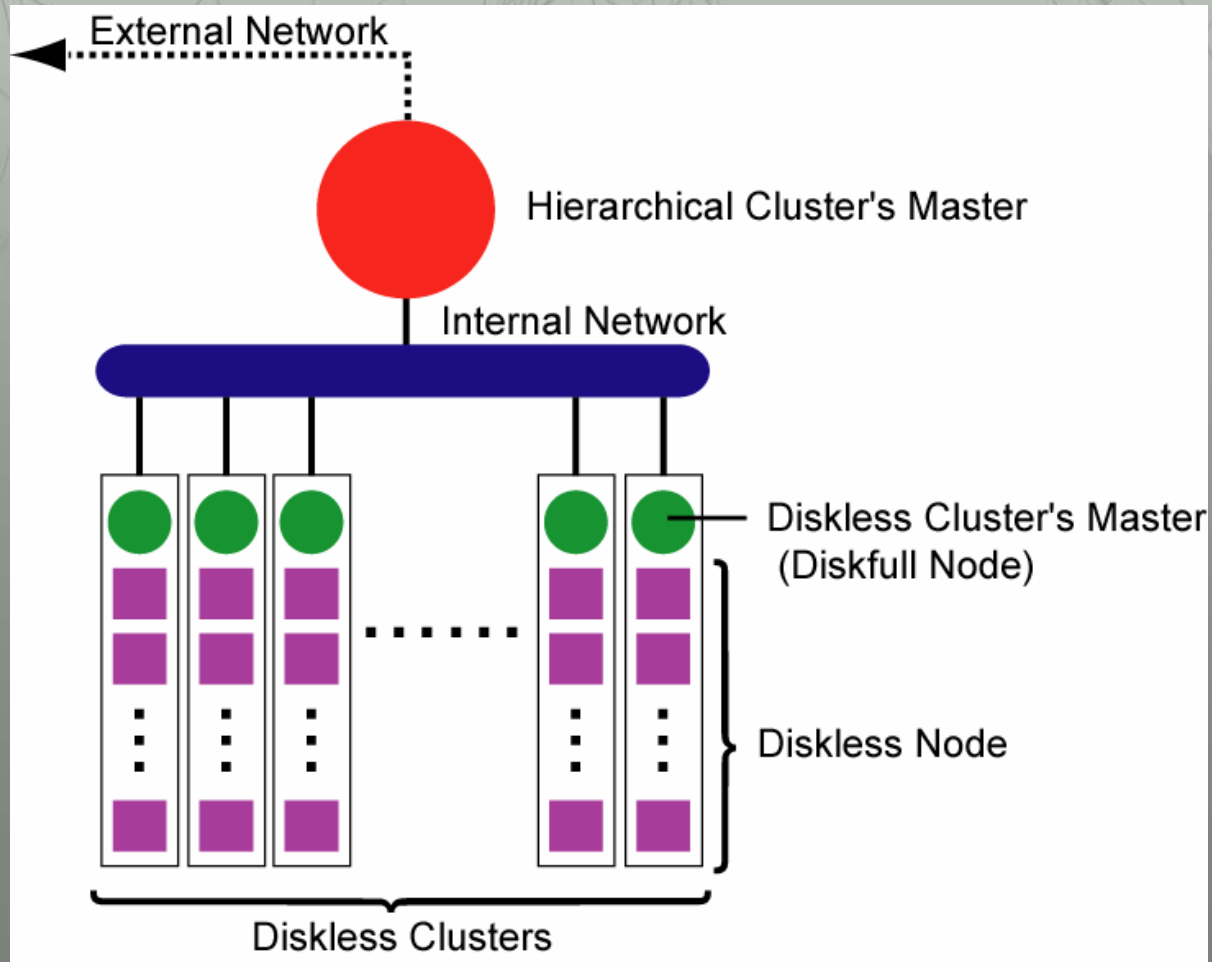
```
#Enter PARTITION size.
FPRT /dev/hda1 128 boot *
SPRT /dev/hda2 512 swap
TPRT /dev/hda3 - /
4PRT none - none
NISDOMAIN nis.org
LOCALETHCARD eth0
# NETWORK NETMASK BROADCAST
NET 192.168.0.1 255.255.255.0 192.168.0.255
#DCASTMASTER Master's name Master's IP
DCASTMASTER mixed01 192.168.0.11
NFSMASTER mixed01 192.168.0.11
GATEWAY 192.168.0.11
#slave's name slave's IP slave's MACaddress
#Autogenerated by update-cluster
mixed02 192.168.0.12 009027D0A80B
mixed03 192.168.0.13 004005A06C67
mixed04 192.168.0.14 004005A886A5
mixed05 192.168.0.15 004005A06427
mixed06 192.168.0.16 004005A40DEE
mixed07 192.168.0.17 004005A40D37
mixed08 192.168.0.18 004005A40D75
mixed09 192.168.0.19 004005A40D0E
#End update-cluster
```

# Procedures (3)

---

Step 5. Invoke dcast-setup. After this process finishes, all slave nodes are rebooted. When all slave nodes finish the process of rebooting, PC cluster construction completes.

# Hierarchical cluster



# Procedures (2)

Step 1. Install Debian GNU/Linux on the top master node.

Step 2. Install DCAST.

Step 3. Creating grub floppy disk.

Step 4. Create slave.lst for diskless master servers.

Step 5. Invoke dcast-setup. After that, boot up all remaining nodes of the diskfull cluster.

Step 6. Create slave.lst for diskless slave nodes.

Step 7. Invoke dcast-setup. After that, boot up all diskless nodes.

# Installation time

256 node cluster

Hierarchical cluster (1 diskfull node + 15 diskless nodes ) \* 16

At top node:

Configuration time: 10 min

making directory for each diskfull node:  $8 * 15 = 120$  min

At each node:

Configuration time: 10 min

making directory for each diskless node:  $4 * 15 = 60$  min

Total: 200 min

# Conclusions

We proposed **Doshisha Cluster Auto Setup Tool (DCAST)**.

DCAST was designed with the following goals:

- Target at novices without specific knowledge to PC clusters
- Avoid interactive operations, and used Debian GNU/Linux for ease of software upgrades
- Software upgrading is done in the same manner as initial install
- Allow creation of both diskless cluster and diskfull cluster
- Combine existing software to create as much of the final product

The procedure to create a PC cluster using DCAST was shown, and the method to maintain the PC cluster after creation using DCAST was described. The operation of master node and slave nodes when creating PC cluster using DCAST was described. Also, the operation and method of adding module scripts for arbitrary software installation was explained.

# In the future ...

In the future, the following enhancements are considered:

- Improve the speed in which root le system template is copied from mater node to slave node
- Addition of more module scripts
- A cluster description markup language for more verbose configuration that allows creation of a hierarchical PC cluster with one dcast-setup invocation