Linux Cluster Management

Stefano Martinelli
(System Management Group)
CINECA

Overview

- Linux Cluster general Analysis
- Management Analysis
- Management in practice

Planning & analizing

Good Management begins with

PLANNING and ANALIZING

These are perhaps the most BORING but IMPORTANT and TIME SAVING activities

Planning to build a cluster/1

What kind of application?

Restricting to HPC applications:
- serial and/or parallel applications
- one or few specific applications / generic uncontrollable applications
- I/O bound and/or CPU bound applications

Planning to build a cluster/2

What kind of load?

- a couple of small cuncurrent application running => Small installation
  You can use usual management tools + some tricks and shell scripts

- various cuncurrent parallel applications running => Large installation
  You NEED efficient and flexible management tools if you want to survive!

Planning to build a cluster/3

Who will use the cluster?
Opposite ends:

- You and your friends => Private environment
  You can:
  - Easily control (near and few) users
  - Easily explain complex usage tricks
- External users => Public environment
  You need:
  - Strict control over (many and unknown) users
  - Easy environment usage: not all users can be very skilled or can be aware of tricks for use (and you don't want to receive hundreds of e-mails!!)

Planning to build a cluster/4

What are the users environmental requirements?

Opposite ends:

- Just testing and experimental activity
  - Rebuilding and experimenting allowed
- Production (possibly 24x7)
  - Reliable and flexible environment
  - Reduced downtime
  - Backup-recovery procedures

Our target cluster

Building a

- HPC mixed (most parallel)
- Large
- Public
- Production

Linux Cluster

Let's start!!/1

step 1: Let's build a Large HPC parallel cluster

A) Where we come from
- ✓ examine previuosly existent parallel HPC architectures
- ✓ see differences with respect to our Linux Cluster
- ✓ try to take the good and bring it to our cluster architecture.

B) Cluster Function general Analysis
- ✓ Point out the functionalities we have to realize in a cluster, and put them together

C) Cluster Management Software Analysis
- ✓ Characterize its general building components and structure
- ✓ Outline its installation steps

Let's start!!/2

step 2: Let's set up the environment for Public users

A) Base environment:
- ✓ Env (vars, modules)
- ✓ mpi

B) Public environment:
- ✓ User control
- ✓ Access control
- ✓ Resource and workload management

C) Production environment
- ✓ Accounting (OS and batch)
- ✓ Recovery procedures: backup, system dumps (of all devices!)

Step 1:A) Where we come from

Possible machine architectures for HPC:

Single System Image
- MPP - Massive Parallel Processors (e.g. Cray T3E)
- SMP - Shared Memory Processors (e.g. SGI Origin)

Cluster
- Distributed cluster of homogeneous single CPU or small SMP nodes (e.g. IBM SP2)
- Distributed cluster of generic heterogeneous machines (e.g. www.distributed.net)

Our Beowulf: Linux Cluster of (nearly) homogeneous (possibly) SMPs

Single System Image vs Cluster/1

Single Image is easier both for users and for system management
- You perform all tasks (compilation, environment setup,program execution) on the same system
- You just have one system (config. files, filesystems) to manage with.

Single System Image vs Cluster/2

Cluster can be better for some management tasks:

- You can isolate interactive activity from parallel/serial batch production (important also for reproducible timing results)
- Some HA mechanisms can be set up

=> hardware and software upgrade can be performed with NO downtime at all!

Cluster towards SSI

We want to take the good from SSI and bring it to cluster environment

$\Rightarrow$ SSI for cluster translates in

"Single Point of Access and Control System"

This drives many successive choices

Step 1:B) Cluster function general Analysis

These Cluster Functions must be carried out by one or more nodes:

- Installation/update
- Management and control
- Users' login
- Compute
- I/O

Cluster Function realization/1

How to realize some functionality can depend on:

- Actually available hardware and software technologies (particularly true for File Systems)
- The relative importance you give to all variables
- Available budget

Cluster Function realization/2

For each function you can:

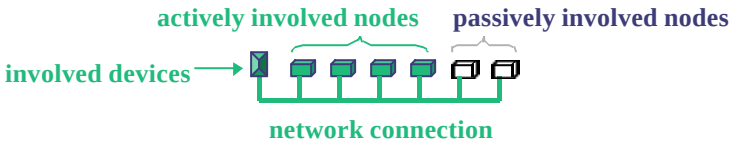
- Describe it
- Establish the infrastructure used to realize it:
  - ✓ Involved nodes and their features
  - ✓ Network connection between nodes and related devices

Cluster Function realization/3

Let's play a little game with:

- ✓ Colors : one for each function
- ✓ Bricks : one for each node involved in the function
- ✓ Diamonds : representing infrastructure devices
- ✓ Lines : representing network connections

For example:
If function "A" is GREEN :



Installation/update function

What's for:

- Responsible for the installation of all the system.
- Exports FS containing OS, libs and software to other nodes
- Provide services required for installation such as DHCP/BOOTP, TFTP, NFS etc.

Nodes:

- One needed

Connection:

- Gather all nodes
- Has only spot use
- Outgoing link from installation node could be strengthen

Management and Control function /1

What's for:

- Manage cluster components
  - ✓Control/configure all devices (nodes, switches, terminal servers)
  - ✓Collect logs/SNMP alarms
- Offer centralized services for normal activity
  - ✓Store Cluster DB
  - ✓DNS , NTP, etc.
  - ✓Batch server and scheduler

This function is vital and critical for cluster activity

Management and Control function /2

Nodes:

- As Redundant (HA,RAID 5 for storage) and Secure (not accessible by users?!) as possible
- No particular CPU or memory requirements

Connection:

- Gather all nodes and devices
- Medium load, but link from management node can be strengthen

Users' login function

What's for:

- Allow the users access the cluster for:
  - ✓ Preparing programs (compilation)
  - ✓ Storing input (and keeping output) data
  - ✓ Using the "back-end" of the cluster, directly or by means of a batch system

Nodes:

- One or more, according to the number of users
- Robust and secure
- Load is nearly unpredictable and difficult to control => do not overlap with batch parallel activity

Connection:

- To the outside world

Compute function

What's for:
- Just to perform calculus

Node features:
- Many
- CPUs, memory size, memory speed and PCI bus speed according to you applications (but consider system activity!)
- High CPU load and few MB of memory available for system

Connection:
- Connect all computation nodes
- Network Latency and Bandwith proportional to IPC features

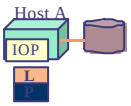
I/O function /1

Definitions

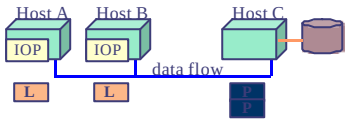
Logical FS visibility	Real FS location
•*Private*: only one node need to see	•*Local*: only one node have a copy of it
•*Global*: all nodes need to see the same data contents	•*Shared*: all nodes see the same copy

Logical/Physical I/O:
- Logical: done by host where starting I/O process (IOP) run
  - ✓ Use resources anyway
  - ✓ Send data to some "entity"
- Physical: done by host writing to final I/O device

1) ex. of L/P I/O by the same host



2) ex. of L/P I/O by different hosts



I/O function /2

Consequences:

⇒ Splitting L/P I/O introduces some I/O network communication between hosts

⇒ Global FS introduces some network communication among hosts and some I/O network data stream among hosts and storage devices

I/O function /3

What's for:

- Storage I/O activity for:
  - ✓System
    - ▪ private (e.g. logs)
    - ▪ global (e.g. products)
  - ✓User
    - ▪ Home (reliability): global
    - ▪ Scratch (performance):
      - • private
      - • global

Private Local FS, one for each node:

Nodes: All nodes

Connection: Not needed!

Global Shared FS => The cluster global FS problem

Nodes: All involved, but some may have special "server" functions

Connection: Among all nodes and, in some way, with storage devices

The Cluster global FS problem

File System Consistency

You want all nodes to see the same files and data
(see previous statements “Single Point of Access and Control”)

Two possible basic strategies:

- Global local: make identical copies of data into many individual FS, one for each node (replication).
  Drawbacks:
  - ✓ Generate network traffic
  - ✓ Waste space
  - ✓ Fragment disk space
- Global shared: only one copy of data is available for all nodes.
  Drawbacks:
  - ✓ Performance
  - ✓ Technological problem: software availability, reliability and complexity

Global FS Technology: Terms and Concepts

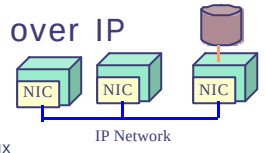
SAN (Storage Area Network)

Network among hosts and storage devices (disks,tapes) devoted to data stream

- ✓ SCSI protocol over FC protocol
- ✓ 3 topologies: point-to-point,Arbitrated Loop, Switched
- ✓ Device-sharing is possible!!
- ✓ 100 MB/s
- ✓ Expensive infrastructure

NBD (Network Block Device)

Low-level export of disk space over IP



Global shared FS Technology for Linux

Old technologies are available:

- NFS (Network FS)
- OpenAFS (Andrew FS)/CODA

New technologies (also exploiting new hardware technologies) are now available:

- OpenGFS/Sistina GFS (Global FS)
- IBM GPFS (General Parallel FS)
- PVFS (Parallel Virtual FS)

Near to come:

- CXFS (now available only for SGI IRIX)

Global FS Technology / 1

NFS

Based on a client-server model

Pros:

- Very well known
- Available for all platforms

Cons:

- Does NOT scale if performance is needed
- Has single point of failure (unless HA implemented)

Global FS Technology / 2

OpenAFS/CODA

CODA is essentially AFS' successor.
Both are more concerned on "distribution" over WAN than on performance over LAN
Original project aim: provide a FS with resilience to network failures
Base on client/server model:

- many clients possible
- Read/write replication servers possible for higher availability
- If disconnected from servers, client can cache write operations and resyncronize when reconnected

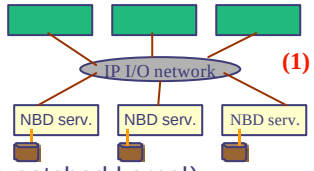
=> They do not appear to be particularly suitable for our filesystems

Global FS Technology / 3

Sistina GFS

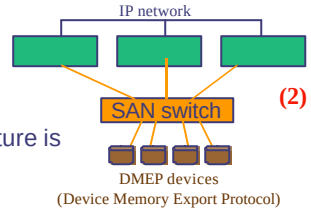
2 ways + hybrids:

1. Use NBD (through TCP network)
2. Use SAN Storage (DMEP devices)



Pros:

- Can be installed as rpm!! (or download a patched kernel)
- Can boot from GFS (CDSL- Context Dependent Symbolic Links)
- Journaled; can be expanded on-line
- Scales from 1 to ?? Nodes (30 tested)



Cons:

- Not GPL any more
- The SAN solution is preferred, but infrastructure is expensive
- Quite new: reliability to be tested

Global FS Technology / 4

OpenGFS

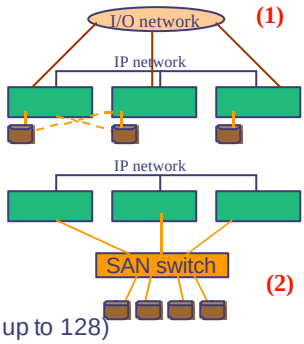
- Has the same general structure of Sistina GFS
- Branch from Sistina GFS started some months ago
- Looking for help!! (http://www.opengfs.org)

Global FS Technology / 5

IBM GPFS

2 unmixable ways of work:

1. Disks belongs to one node (and maybe one backup node); remote disk access performed throught an I/O network
2. Disks can be directly accessed by all nodes



Pros:

- Install as simple rpm!
- Use SCSI/EIDE disks/partitions
- Well tested in SP and cluster AIX environments
- Journaled FS; can be expanded on-line

Cons:

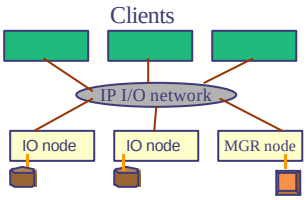
- Supports only >= RedHat 7.1 kernels
- Not free (and not cheap!) but you have support!
- v1.1.1 Scale up to 32 nodes (v1.2 should scale up to 128)

Global FS Technology / 6

PVFS

Primary goal: provide high performance “global scratch space”
Has 3 kind of nodes: clients, IO servers, and one metadata server
Pros:

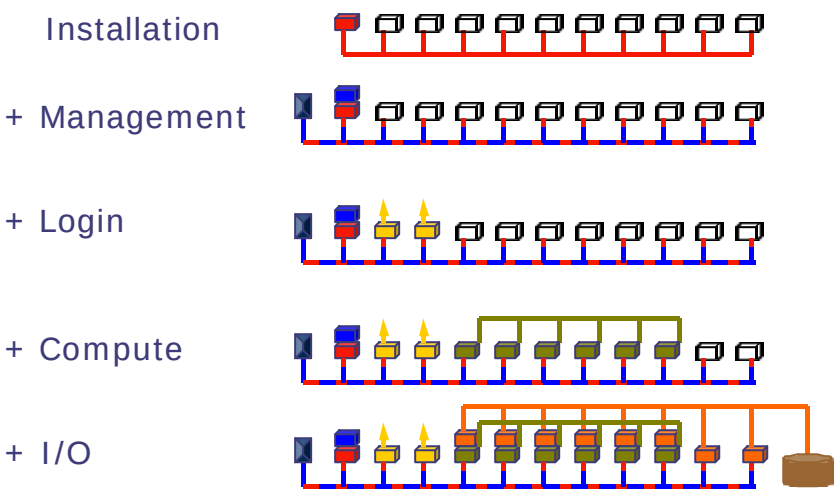
- Quite simple to install
- Use watever filesystem on server IO nodes
- Based on client/server daemons (pvfsd/iod)
- Provide transparent access and specific APIs
- No kernel support limitation (apparently)
- No kernel patch, but kernel module on client
- Free!!!
- Seems to scale very well (tried on 256 dual P III cluster)



Cons:

- cannot be expanded on-line ?
- More concerned on performance rather than on reliability
- Administration not very much flexible during run time

Putting functions together



Our "Simplified" Cluster model

Many cluster structures are possible.
Management can be different depending on the structure.
Our cluster basic hardware and software model:

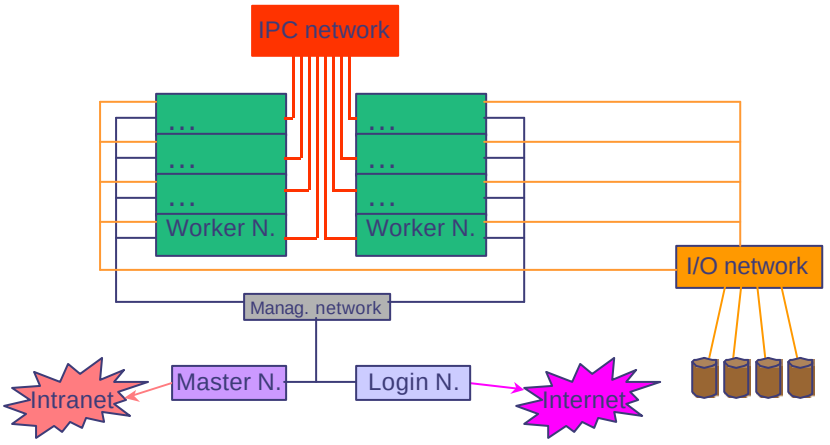
Node Names	Functions included
One Master Node (MN)	Administration Installation Management
One Login Node (LN)	User login Interactive activity Compilation
Many Worker Nodes (WN)	Computation and/or I/O

All nodes have local disks and OS installed.

Cluster connections

Network	Function	Connected elements
Internal private tcp/ip network	Administration and some little I/O	nodes, switches ,etc
IPC network	Fast & low latency communication (I/O,MPI)	compute nodes
Internet connection	Internet connection (login,ftp,backup)	Login node
SAN (Storage Area Network)	Connection with storage devices (disks,tape drives)	Some/All cluster nodes, FC switch, storage

Cluster Components



Ready for cluster installation!!

We have ended up defining our cluster:

- its components
- functions of its components

How to setup these components so that they can accomplish their function?

This is what the "Cluster Management Software" is for.

Step 1:C) Cluster management software Analysis

What do we expect from our cms?

Functional Areas	Desired features
Cluster nodes Installation/updates	•All required services provided •No manual intervention needed (full remote control) •Heterogeneous nodes allowed (node customization)
Management and control	•Cluster wide commands •Everyday services provided •Monitoring tools •Security and access control mechanisms (user management) •Workload Management System
Programming User Environment	•Parallel paradigm support (MPI,PVM)

Suggestions to ease management and decrease errors:

- Do NOT duplicate configuration DATA when possible

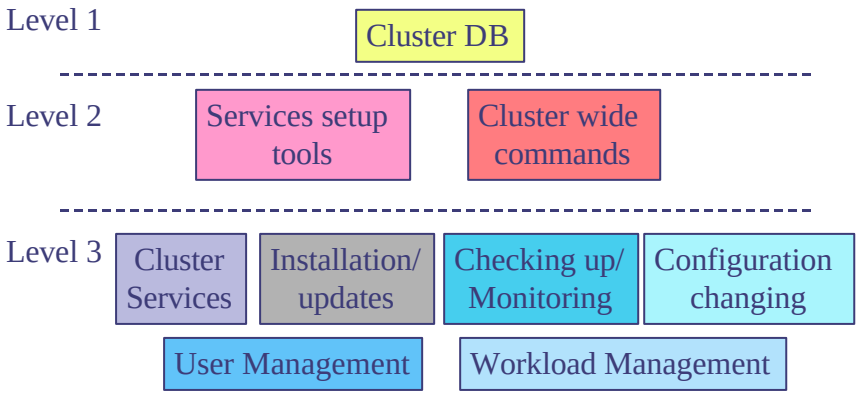
=> *use central repository*

- Less manual activities as possible

=> *use scripts to perform atomic operations*

Cluster Management Software /1

General structure and components:



Most Popular Cluster Management Software available

Free and Open:

- OSCAR (OpenSource Cluster Application Resources)
- NPACI Rocks
- xCAT (xSeries Cluster Administration Tools)

Commercial:

- IBM CSM (Cluster Systems Management)
- Scyld Beowulf

Are generally made of an ensamble of already available software packages thought for specific tasks, but configured to operate together, plus some add-on.

OSCAR

Complete and easy clustering solution: eliminate download & install/configure of individual components

Features:

- LUI for cluster installation
- PXE or etherboot for node boots
- C3 for cluster wide commands, file distribution, remote shutdown
- OpenPBS-MAUI for batch system
- RedHat 7.1, but not only! (distribution used must support RPM installation packages)
- 3 installation levels (from “simple” to “expert”)
- Precompiled packages

NPACI Rocks

Goal: make clusters easy to deploy, manage, upgrade, scale

Features:

- Based on kickstart and RPM (RedHat)
- Use MySQL for DB
- Hardware neutral
- Heterogeneous nodes easily allowed
- Network-enabled Power Disatribution Unit (PDU) => nodes reinstall at every hard power cycle
- NIS for account syncronization
- NFS for HOMEs
- PBS+MAUI

xCAT

Based on IBM x330 nodes (specific hardware features such as PXE nics, ASMA, SP) and some other hardware, but can essentially be used with any other x86 system

Features:

- Based on RedHat, kickstart and RPM
- No wizard => follow instructions step by step
- Many packages have to be compiled, but scripts are present for preparation => you can have full control of ALL and perform “heavy” customization
- Great installation method (PXE and PXElinux)
- OpenPBS+MAUI

IBM CMS

Derived from IBM PSSP (Parallel System Support Programs) for SP2 system management

Features:

- Supported only on IBM Cluster 1300 Model (bunch of x330 plus some other hardware)
- Based on RedHat 7.1, RPM and kickstart
- Different branch from xCAT, but basic functionality is the same (which will survive?)
- Code to be installed in two parts:
  - ✓ Management node part
  - ✓ Cluster node part
- Not free

Scyld Beowulf

True “single system image”:
- Only one installation (of master node)
- Cluster nodes are “minimally” installed at each boot
- Modified 2.2.19 kernel (2.4 porting planned) for clustering operating system => transparent process migration towards cluster nodes
- Use PVFS for Global FS
- Have their own batch system, but support PBS

Pros:
- Functional production system tested

Cons:
- Closed system:
  - ✓kernel cannot changed by you
  - ✓Bug fixes and changes must be applied and tested by Scyld
- Cannot exploit some cluster benefits
- Limited support for heterogeneous nodes
- Do NOT deal with our cluster model

Cluster Installation Steps

- Auxiliary devices configuration
- Master node installation
- Cluster management software installation
  - ✓Cluster Management Software basic installation
  - ✓Cluster DB setup
  - ✓Cluster Management Software setup
- Cluster nodes installation
  - ✓Network installation
  - ✓Software installation
    - Local
    - Shared
- Post Client Install Cluster Configuration
- Installation checkup

Auxiliary devices configuration

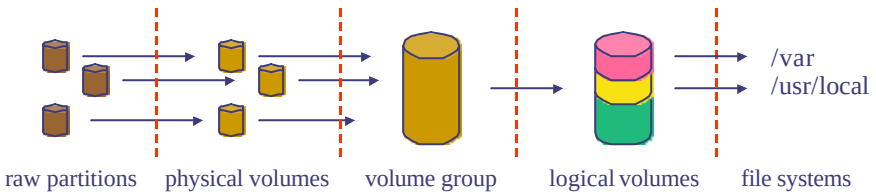
- Switches of different kind of networks:
  - Ethernet switches -> vlans
  - SAN switches -> zoning, if necessary
- Storage LUNs allocation on storage devices
- Remote control devices:
  - Network-enabled PDUs
  - Terminal servers
  - ASMA cards

PLEASE CHANGE THEIR DEFAULT LOGIN PASSWORD!

Master Node installation

Usually performed as a simple workstation

- Only once at the beginning !!(the whole cluster depend on it)
- Include some standard “server” packages
- Some cms provide CD iso images with customized distribution
- Use LVM (Logical Volume Manager) for FS if possible
  - ✓ Installed as a kernel patch
  - ✓ Let you resize FS
  - ✓ Let you aggregate disks/partitions



Cluster management Software basic installation

- Check/perform installation of some required “standard” services and/or packages.
  - ✓ Usual services included:
    - Tftpd
    - Dhcp/bootp
    - DNS (named)
    - Nfs
  - ✓ Usual software included:
    - Sources for nodes installation in appropriate FS
- Install Cluster Software scripts

Cluster DB setup

Describe your cluster components in appropriate files (can be a guided process).Usually:

- Hostname, IP and MAC address for each node
- Connection properties and description
- Choice of remote communication programs
- FS partitions
- kernel version
- Etc……..

Cluster management software setup

Based on data inserted in cluster DB, some actions may be performed in order to complete the cms installation:

- further software download
- boot floppy disks preparation
- services config files setup (dhcp,DNS)
- etc..

Note: OSCAR has wonderful step-by-step Wizard for these steps!

Cluster nodes installation/1

Different possibilities:

1) local boot + local installation (using local devices - CDs)
   - ✓ Extremely human-time long and difficul customization
2) local boot + network installation
   - ✓ Needs boot disk setup (floppy or CD), needs NFS/TFTP/HTTP server setup (but only once!) but easier node customization
3) network boot + network installation
   - ✓ Needs network-boot-enabled card, needs NFS/TFTP/HTTP server setup (but only once!) but easier node customization

2) and 3) are quite similar, and projects such as the "etherboot" project make them nearly equal

Cluster nodes installation/2

All installations can be performed:

- interactively
- non-interactively

Non-interactive installations:

- let you save a lot of human time
- are less error prone.
- are performed using programs (such as RedHat Kickstart or SuSe ALICE) which
  - ✓ "simulate" the interactive answering
  - ✓ can perform some post-installation procedures for customization
- Are better performed using DHCP => you can use the same config files (or installation CDs) for all the installations

In conclusion: network non-interactive installation is better!!

Cluster nodes network installation

Network booting solutions:

PXE (Preboot eXecution Environment)

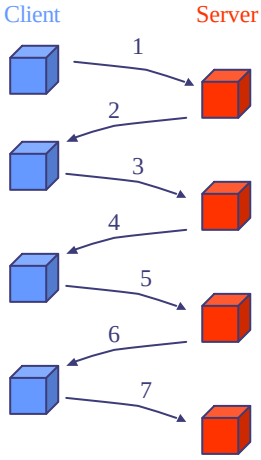
- Open standard environment proposed by Intel for the BIOS of motherboards with integrated net interfaces:
  - ✓ Motherboards should be able to boot from the network following the BOOTP/DHCP protocol, with some particular extentions
  - ✓ The BIOS should present some BOOTP/DHCP APIs to the upper program execution environment.

“Etherboot” project

- Project for letting a system boot off the network, after a first preboot from a local device (floppy or CD)

Network Installation dynamics

1. Client start booting in dhcp (from BIOS or from floppy)
2. Server return network config and network path where to find (usually with TFTP) the executable to boot with
3. Client contact the TFTP server, asking the boot executable
4. Server return boot executable
5. After partitioning disk & creating file systems, Client contact the server asking with some protocol (NFS,TFTP or HTTP) all necessary software and files for installation
6. Server return all software and files required
7. After software installation, Client execute some post-install script.Eventually,at the end, the client communicate to the server that he has terminated installation, then goes to prompt or reboots



Software installation / 1

Some software has:

- Some part private to nodes with the same path (e.g. /var/spool/pbs for PBS) => can be local
- Some other global part consisting of identical programs and data, and with small access frequency (e.g. main PBS commands, mpich commands, PGI compilers)=> replicating or sharing?

Software installation / 2

Global files: Replicating vs sharing

Replicate if files:

- Are relative to OS (passwd file, kernel modules)
- Are not usually changed
- Do not occupy too much space
- Could be customized differently for each node in the future!

Sync mechanisms (rdist, rsync etc.) can be used

Sharing is generally better for:

- Absolute consistency
- No waist of disk space

but consider:

- Technology used
- I/O amount

Cluster nodes FS structure

General FS tree organization after installation:

Local FS:

/	file system root
/etc	configuration files
/bin,/usr/bin	most user commands
/usr/lib	system libraries
/tmp	system tmp
/scratch	user tmp (suggested)

Shared FS:

/home	home directories
/usr/local	libraries, application packages, software, scripts and config files shared by all nodes
/SCRATCH	global user tmp

Post Client Install Cluster Configuration

Try to reduce this phase, because it has to be executed in general after each node installation. Some of these activities can be:

- OpenSSL/SSH setup for Cluster wide commands functionality => remote command execution on nodes without giving passwords
- Installation of packages which require the system to be up and running (e.g. GPFS)
- Etc..

Installation Checkup

Scripts/progs for testing basic functionality
Some will have to be written by your own, but they are worth => they let you save a lot of troubleshooting time
Usual tests include:

- System consistency
- Correct driver installation tests
- PVM/MPI tests

Note as examples OSCAR Cluster Tests!

Execute them after each node installation!!!!

So, after a successful installation…

Management and control

day-by-day activities:

- Cluster services
- Node Configuration changes
- Monitoring
- User management
- Workload management

Many of these activities can be better performed using cwc

Cluster wide commands /1

We want something like a prompt from which we give shell commands concurrently to every node, so that we emulate a “single system image” prompt.

Eg. :

```
[master]# global_sh
global> whoami
node1: root
node2: root
node3: root
node4: root
...
nodeN: root
global>
```

Cluster wide commands /2

Usually, cwc need:
- remote access mechanisms to cluster nodes without giving a password

Usually, cwc provide at least:
- Concurrent remote execution
- Concurrent remote file copy

- Cwc are useful espetially on homogeneous environments
- Every cms has some cwc functionality
- Independent projects are:
  - the C3 project
  - “pconsole” utility

Cluster services

Some standard services are:
- NFS
- DNS
- Time sincronization (NTPD)
- Config files syncronization through replication (rdist,rsync)

Node Configuration changes

Changes can be performed adding/upgrading software or changing config files. Actions depend on “localization” of files added/changed:

- Shared files => do only once
- Replicated files, which can be:
  - ✓ Periodically kept in sync through authomatic distribution => do only once
  - ✓ No kept in sync => do only once using cluster wide commands to propagate files and actions AND update installation procedure in order to include these changes

Monitoring Tools / 1

We presumibly want to be warned if:

- A node hangs
- A node is unreachable due to a kernel or network problem
- Pbs daemons die on master or cluster nodes
- A FS is full

Available tools:

- Provided by cluster management software
- Available as stand-alone tools: take your preferred one:
  - ✓ Netsaint
  - ✓ Big Brother
  - ✓ MON
  - ✓ Net-SNMP
  - ✓ many many others...

Monitoring Tools /2

In general:

- Are made of scripts (shell, perl, etc.)
- Use DB text files
- Have “plugins”, so checks can be customized simply writing a plugin script or program
- Generally work by polling nodes at fixed intervals using “plugins”
- Many have web interface

Always good practice: redirect syslog messages to master node for easier debugging!

Step 2:A) Basic user environment

Let users enter into Login node using ssh (Secure Shell) which implements encripted communication:

- OpenSSH is free implementation (of both protocol 1 and 2)
- Offers encripted login, but also :
  - ✓ Remote copy (scp)
  - ✓ Remote execution (ssh)
  - ✓ ftp with encription (sftp)
- Use PAM-mified version for greater flexibility (see later)

User accounts must be defined also on all compute nodes => use NIS (security risks!) or passwd files replication

The user environment/1

Shell variables set by system (of all nodes!!!) in
- /etc/profile
- /etc/csh.login

and consider files in /etc/profile.d/

Shell variables set by users in users' profile files:
- $HOME/.bash_profile
- $HOME/.bashrc

For new users: modify prototype profile files in /etc/skel/

The user environment/2

Modules (NOT kernel modules!)
- Not only for cluster environments.
- Different "prepared" user environments, with setting of variable sets
- Very useful for keeping different versions of the same product, or different similar products (e.g. compilers)
- Users do not need to be aware of where products are

Usage:

Source an appropriate file for your shell, then:

```
# module list
# module load
# module avail
```

MPI user environment

Use mpich, or LAM/MPI, which are free implementations of the MPI protocol standard

Mpich_gm: mpich version using the Myrinet driver (download from Myricom web site)

Undesired features:

- Users have to prepare some “hosts” files, indicating the hosts they want to use for their parallel applications
- Users have to be able to access all nodes where their tasks run
- Many users could run on the same node many different tasks, without knowledge of each other

=> A resource manager could help!!!

Step 2:B) Public user environment

User control

Quotas

- Used usually on HOME FS
- Easy to use (but kernel must be enabled!)
- Package “quota” needed
  - ✓Put “userquota” in /etc/fstab mount options
  - ✓# quotaon *filesystem*
  - ✓# quotacheck *filesystem*
  - ✓# edquota [-p prototype_user] user

CPU/memory Limits on login node

- Use PAM-mified ssh and the “pam_limits.so” module

PAM /1
(Pluggable Authentication Module)

Idea is simple and great: separate applications from authentication-authorization mechanisms:

- All PAM-mified applications use PAM library functions for authentication-authorization
- PAM library functions look into the PAM config modules and files, which are text files (usually in /etc/pam.d/)

Result: authentication-authorization mechanisms can be changed from a central point, without changing the applications!

PAM is:

- Very powerful and flexible system
- Neither too much complex nor too simple!

PAM /2

Applying PAM for CPU/memory limits control

File /etc/pam.d/sshd contains a “session” entry with the “pam_limits.so” module:

```
auth       required     /lib/security/pam_pwdb.so shadow nodelay
auth       required     /lib/security/pam_nologin.so
account    required      /lib/security/pam_pwdb.so
password   required       /lib/security/pam_cracklib.so
password   required       /lib/security/pam_pwdb.so shadow nullok
   use_authtok
session    required      /lib/security/pam_pwdb.so
session    required      /lib/security/pam_limits.so
```

PAM /3

pam_limits.so module parses the /etc/security/limits file, and set user limits accordingly.
Here is an example of /etc/security/limits file:

```
#*              soft    core            0
#@student       hard    nproc           20
#@faculty       hard    nproc           50
#*              hard    data            10000
#*              hard    rss             10000
#*              hard    memlock         10000
#*              hard    stack           10000
*               hard    cpu             10
*               soft    nproc           1000
*               hard    maxlogins       100
root            -
```


Access control /1

On login node: usually no access control
On compute nodes: NO default login access by users => only controlled applications can be allowed to run

You can control users' access through ssh using PAM

Access control /2

To control users' access to nodes using ssh, add this entry at the end of /etc/pam.d/sshd:

```
account    required     /lib/security/pam_access.so
```

Module pam_access.so uses file /etc/security/access.conf for access control settings. To **forbid** access to all users but root, access.conf can be as follows:

```
# Format of the login access control table is three fields separated by a
# ":" character:
#
#      permission : users : origins
#
# The first field should be a "+" (access granted) or "-" (access denied)
# character.
#
-:ALL EXCEPT root:ALL
```


Access control /3

We have forbidden access to nodes. However,how MPI applications can be executed?

Mpich uses either rsh or ssh for remote task execution!!!

⇒Always forbidding is not enough

⇒We want to grant access to some users during a certain time window

A resource manager is needed!!!

Programme execution

Two ways:

- Interactive: users' programs are directly executed by the login shell, and run immediately
- Batch: users' programs (*jobs*) are submitted to a system program which will execute them in the future, according to the site policy, to best satisfy the site production needs

If we want to have control, we have to follow the "batch" way

A resource and workload manager is needed!!!

Resource and Workload management

Provided by a so called "Batch System"

- Some piece of software control available resources
- Some other piece of software decide which application to execute based on available resources
- Some other pieces of software are devoted to actually execute applications

Resorce Management and Access Control

The batch system knows who will be allowed to run applications on which nodes

⇒some mechanisms can be put in place to allow access to nodes only to these legitimate users

Usually batch system's prologue and epilogue programs can be used for the purpose (eg. modifying access.conf file appropriately)

Batch systems for Linux

There are several batch queumg systems available for Linux-based clusters, depending on what your needs are. Here are just a few:

- Condor (http://www.cs.wisc.edu/condor)
- DQS (http://www.scri.fsu.edu/~pasko/dqs.html)
- Job Manager (http://bond.imm.dtu.dk/jobd/)
- GNU Queue (http://www.gnu.org/software/queue/queue.html)
- LSF (http://www.platform.com, -- commercial)
- Portable Batch System (PBS) (http://www.openpbs.org,http://www.pbspro.com)

PBS Batch system

The *de facto* most widely used is (Open)PBS

- Quite simple
- Thought for clusters
- Open and Free
- A “professional” version is available by Veridian, not free but with support
- PBS provides:
  - Resource manager (pbs_server daemon)
  - Scheduler (pbs_sched daemon)
  - Many executors (pbs_mom and pbs_resmon daemons)

PBS and MAUI scheduler

- The head of the system stands in the scheduler.
- The head can be changed if you have a better one than its default.
- The MAUI scheduler is a very good alternative, implementing the “backfill” algorithm

Step 2:C) Production user environment

Accounting:

- OS
  - ✓ Sysstat package for “sar” statistics
  - ✓ Pacct package for process accounting
  - ✓ Some scripting needed for standard daily and monthly UNIX accounting
  - ✓ Some more scripting is needed for cluster accounting collection
- Batch
  - ✓ Batch systems provide logs for accounting purposes
  - ✓ Sometimes parallel job accounting is difficult

Backup procedures

Goal: Save files to be restored in case of user’s unintentional deletion or disk failure

FS Backup strategies:

- Incremental backup needed for large FS
- In general only of Master node (other nodes can be reinstalled)

Software available:

- Free:
  - ✓ tar/dump on tape device (non-incremental!!)
  - ✓ Arkeia (partially free- only one installation)
- Commercial:
  - ✓ Tivoli TSM
  - ✓ Legato
  - ✓ Veritas

Backup also switches and other devices configurations!

Recovery procedures

Goal: Save system image snapshot to be completely restored in case of some kind of disaster or system intrusion

Restore strategies:

- In general only of Master node
- Made during run-time if possible
- Saved eventually on CD support for immediate reinstallation

Software available:

- Free:
  - ✓ Dump
  - ✓ Partition Image (but FS must be off-line!!)
- Commercial
  - ✓ Norton Ghost

References /1

OpenAFS: http://www.openafs.org
OpenGFS: http://www.opengfs.org
GFS: http://www.sistina.com
GPFS: http://www-1.ibm.com/servers/eserver/clusters/software/gpfs.html
Netsaint: http://www.netsaint.org
Big Brother: http://bb4.com
MON: http://www.kernel.org/software/mon/
NET-SNMP: http://net-snmp.sourceforge.net
OSCAR: http://www.openclustergroup.org
xCAT: ftp://www.redbooks.ibm.com/redbooks/SG246041/
Rocks: http://rocks.npaci.edu
CMS: http://
Scyld: http://www.scyld.com
Webmin: http://webmin.com/webmin
OpenPBS: http://www.openpbs.org
MAUI: http://www.supercluster.org
NBD: http://www.it.uc3m.es/~ptb/nbd/
PVFS: http://www.parl.clemson.edu/pvfs/
CODA: http://www.coda.cs.cmu.edu/
Modules: www.go.dlr.de/fresh/linux/src/modules-3.1.5.tar.gz
SAN: http://www.brocade.com/SAN/white_papers.jhtml
Pconsole: http://www.heiho.net/pconsole/
Arkeia: http://www.arkeia.com/
Partition Image: http://www.partimage.org/
Tivoli TSM: http://www.tivoli.com
Legato: http://www.legato.com
Veritas: http://www.veritas.com

References /2

LUI: http://oss.software.ibm.com/lui

SIS: http://www.sisuite.org

MPICH: http://www-unix.mcs.anl.gov/mpi/mpich

LAM/MPI: http://www.lam-mpi.org

PVM: http://www.csm.ornl.gov/pvm

OpenSSL: http://www.openssl.org

OpenSSH: http://www.openssh.org

C3: http://www.csm.ornl.gov/torc/C3

SystemImager: http://systemimager.sourceforge.net

Etherboot: http://etherboot.sourceforge.net/

Norton Ghost: http://www.symantec.com/sabu/ghost/ghost_personal/