Setting Up Clusters

- Software Provisioning Challenges
- There are no homogeneous clusters
- Functional Inhomogeneity
  - Login & computer nodes
  - Specialized service nodes
    - System logging, I/O, login, dedicated installation nodes
- Hardware inhomogeneity
  - New nodes unlikely to be identical
  - Replacement parts may be different
  - Even parts in original machines may differ, even if they have the same part number
    - Ex: SCSI drives 980 & 981 cylinders defeat imaging program

Functional Differentiation

- Arises from need to scale services
- Mid-size cluster node types
  - Head node/Frontend node
  - Computer node
  - I/O server
  - Web server
  - System logging server
  - Installation server
  - Grid gateway node
  - Batch Scheduler and cluster-wide monitoring

System Software Consistency

- Avoid small differences in C libraries
  - Performance and correctness problems
- New nodes must have identical software & configuration
- Diskless clusters avoid the problem by mounting a uniform file system through NFS

Hardware Provisioning Challenges

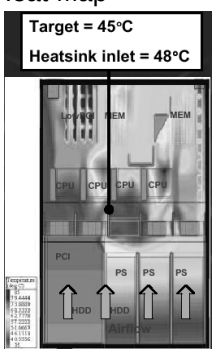
- Organization and labelling can help in debugging problems
- Four areas
  - Node layout – rackmount v workstation tower v blades
  - Cable management
  - Airflow management
  - Power management
- Rack Units
  - 1U = 1.75", standard rack is 2m tall (42U)
  - With higher density (1U) CAP (cable, airflow, power) more important
  - Group cables by tie-wrapping 4 ethernet or 8 power cables – use wire ties every 6 to 12in

Rackmount

- Ethernet cable lengths depend on node types
  - Workstation towers – prebundle 2 each 5,6,7,8'
    - Even on one end for switch
  - 2U rackmount – bank of 8 only 15in high
- Power cables more complicated
  - Need to make sure power cables don't obstruct airflow
  - High-end nodes can dissipate 150-200W
- Need to ensure enough power circuits and distribution units are available
  - Use standard Power Distribution Units (PDUs) rather than power strips
    - Thicker quality cabling which will not overheat

Chassis Heat Map

- Experience with IBM and Compaq
  - And a few white boxes
- Thermal design is important
  - Heat causes premature failures



Power Distribution Units

- Plug power cord from chassis (or towers) into them
- Some units are network addressible
  - Can control outlets via an ethernet network
- USD$400 for 8 outlet PDU

Installation Management

- Open source management systems
  - NPACI Rocks, OSCAR, Score, Scyld, XCAT
- Have to choose range of distributions (RedHat, SuSE, Debian, Mandrake) and hardware supported
- Each distribution has own style, file layout, package formats & definitions, hardware support etc.
- Packaging definitions can cause problems in resolving dependencies
- Linux distributions do hardware detection to install right device drivers
- Hardware getting more diverse

Installation Management

- Diverse hardware
  - Disks (IDE, EIDE, UATA, SATA,SCSI, SAN etc
  - Interconnects (GE, Scali, Myrinet, Quadrics, Infiniband)
  - Motherboards, chipsets, processors
- Cluster building toolkits tend to scale across hardware or distributions but not both

Scaling Choices

- Scaling across distributions
  - Need to make generalizations
  - Take over base installation and hardware detection from distribution
  - Adv: more distribution choice
  - Disadv: a lot of diverse hardware to handle
- Scaling across hardware
  - Use single distribution
  - Leverage built-in installation and hardware detection

Approaches

- Disk Imaging
  - Initially the normal choice for clusters
  - Image based programs: Norton Ghost, PowerQuest Drive Image, SystemImager, Chiba City Imager, PowerCockpit
  - Image based toolkits: OSCAR, Chiba City, CLIC
- Description based installers
  - Use text files to specify files and instructions for configuration
  - Programs: RedHat KickStart, SuSE YaST, Debian FAI
  - Toolkits: NPACI Rocks, IBM XCAT, European Data Grid LCFG
  - Capture disk partitioning, package listing, and software configuration
  - Description can work on many variants of hardware using distribution installer for low level details

Basic High Level Steps

1. Install head node
2. Configure Cluster Services on Head Node
3. Define Configuration of a Compute Node
4. For each compute node – repeat
   a) Detect Ethernet hardware address of new node
   b) Install complete OS on new node
   c) Complete configuration of new node
5. Restart services on head node that are cluster aware (e.g. PBS, Sun Grid Engine)

1./2. Head Node

- OSCAR has user setup configuration separately from installing the toolkit
- NPACI Rocks combines two

3. Define Configuration of a Compute Node

- For disk imaging a *golden node* needs to be configured
  - OSCAR's System Installation Suite (SIS) uses a package list and a set of GUI's to configure this without first installing node
  - Rocks uses a general description which works across hardware types

4. For each compute node

a. Detect Ethernet hardware address of new node
  - On boot uses DHCP
    - Sends MAC address
    - Gets IP, netmask, routing, node name, etc
  - Toolkits have mechanism to detect new MAC addresses
    - Rocks probes /var/log/messages for DHCPDISCOVER requests from new MAC addresses
    - OSCAR uses **tcpdump**

b. Install complete OS on new node
  - Image based: download *golden image,* adjust for disk geometry, IP address etc, and install image
  - Description based: download text based description, and use native installer
    - Pakages are downloaded from a distribution server

4./5. For each compute node

- **Image: most information in golden image**
- **Description: most information in text configuration files**

c. To complete
  - **Used to have to be done explicitly by sysadm**
  - **Now fully automated**

5. Restart services on head node that are cluster aware

NPACI Rocks Toolkit – rocks.npaci.edu

- Techniques *and* software for easy installation, management, monitoring and update of clusters
- Installation
  - Bootable CD + floppy which contains all the packages and site configuration info to bring up an entire cluster
- Management and update philosophies
  - Trivial to completely reinstall any (all) nodes.
  - Nodes are 100% automatically configured
    - Use of DHCP, NIS for configuration
  - Use RedHat's Kickstart to define the set of software that defines a node.
  - All software is delivered in a RedHat Package (RPM)
    - Encapsulate configuration for a package (e.g.. Myrinet)
    - Manage dependencies
  - Never try to figure out if node software is consistent
    - If you ever ask yourself this question, reinstall the node

NPACI Rocks

- Software Repository
  - Red Hat derived distribution
  - Managed with rocks-dist
- Installation Instructions
  - Based on Kickstart
  - Variables in SQL (MySQL)
  - OO Framework used to build configuration/ installation hierarchy
  - Functional decomposition into XML files
    - 100+ nodes
    - 1 graph
  - Python program to convert into Kickstart file (see Fig 6.3)
  - RedHat Anaconda used as installer to interpret Kickstart files

Kickstart

- Describes disk partitioning, package installation, post-configuration (site specific)
- Three sections
  - Command : answers to interactive installation questions
  - Packages: RPMs
  - Post: scripts to configure packages – site specific

Rocks State – Ver. 2.1

- Now tracking Redhat 7.1
  - 2.4 Kernel
  - "Standard Tools" – PBS, MAUI, MPICH, GM, SSH, SSL, ...
  - Could support other distros … don't have staff for this.
- Designed to take "bare hardware" to cluster in a short period of time
  - Linux upgrades are often "forklift-style". Rocks supports this as the default mode of admin
- Bootable CD
  - Kickstart file for Frontend created from Rocks webpage.
  - Use same CD to boot nodes. Automated integration "Legacy Unix config files" derived from mySQL database
- Re-installation (a single *HTTP server*, 100 Mbit)
  - One node: 10 Minutes
  - 32 nodes: 13 Minutes
  - Use multiple HTTP servers + IP-balancing switches for scale

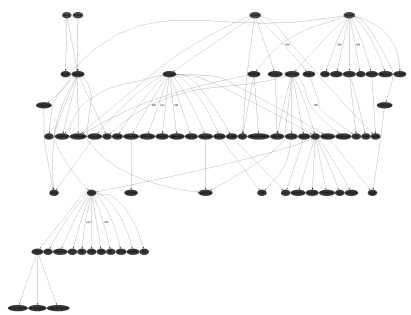
Rocks-dist

- Integrate RedHat Packages from
  - Redhat (mirror) – base distribution + updates
  - Contrib directory
  - Locally produced packages
  - Local contrib (e.g. commerically bought code)
  - Packages from rocks.npaci.edu
- Produces a single updated distribution that resides on front-end
  - Is a RedHat Distribution with patches and updates applied
- Kickstart (RedHat) file is a text description of what's on a node. Rocks automatically produces frontend and node files.
- Different *Kickstart* files and different distribution can co-exist on a front-end to add flexibility in configuring nodes.
- Kickstart files do not contain package versions – Anaconda resolves generic references to package versions

Component Based Configuration

- Rocks used *modules* as building blocks for appliances
  - Small XML files
- A framework describing inheritance is used
  - Directed graph
    - Vertices : configuration of specific service
    - Edges: relationships between services
- When a node is built the kickstart file is generated on-the-fly by traversing the graph
- See 6.5.1 for more details

Kickstart ConfigurationGraph

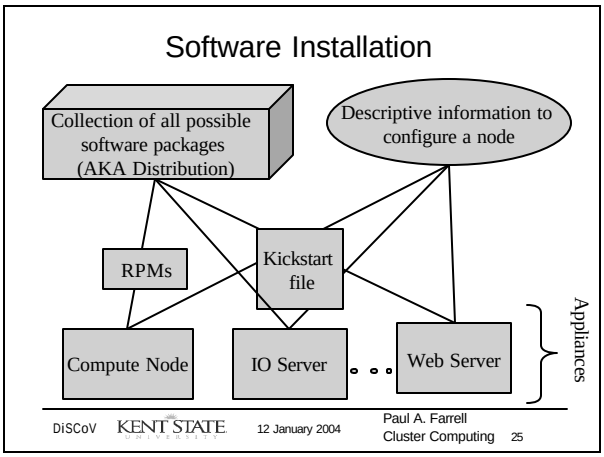


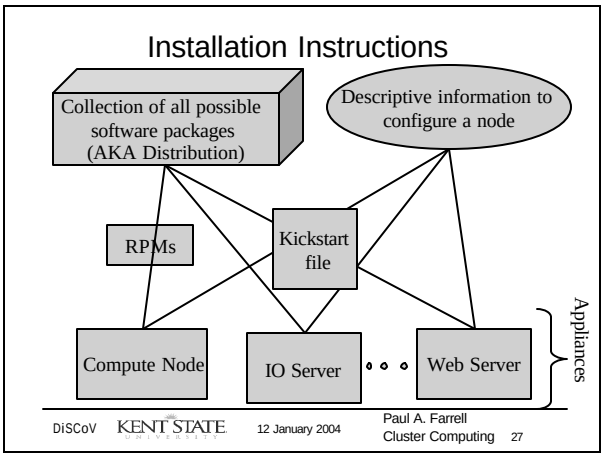
Sample Graph File

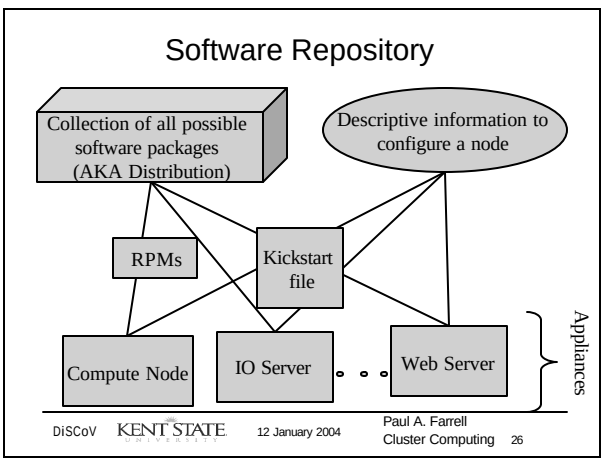
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE kickstart SYSTEM "@GRAPH_DTD@">

<graph>
        <description>
        Default Graph for NPACI Rocks.
        </description>

        <edge from="base" to="scripting"/>
        <edge from="base" to="ssh"/>
        <edge from="base" to="ssl"/>
        <edge from="base" to="file" arch="i386"/>
        <edge from="base" to="lello " arch="ia64"/>
                .
                .
        <edge from="node" to="base" weight="80"/>
        <edge from="node" to="accounting"/>
        <edge from="slavenode" to="node"/>
        <edge from="slavenode" to="nis-client"/>
        <edge from="slavenode" to=" autofs -client"/>
        <edge from="slavenode" to="dhcp -client"/>
        <edge from="slavenode" to="snmp -server"/>
        <edge from="slavenode" to="node -certs"/>
        <edge from="compute" to="slavenode"/>
        <edge from="compute" to="usher -server"/>
        <edge from="masternode" to="node"/>
        <edge from="masternode" to="x11"/>
        <edge from="masternode" to="usherclient"/>
</graph>
```

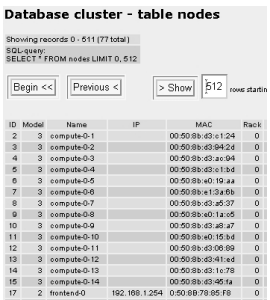




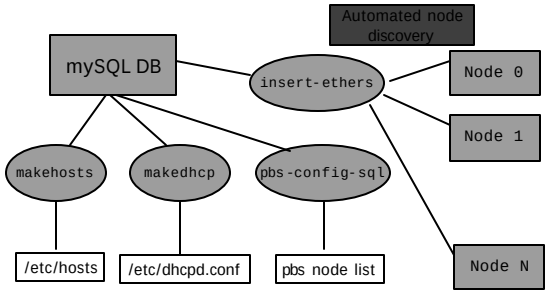


insert-ethers

- Used to populate the "nodes" MySQL table
- Parses a file (e.g., /var/log/messages) for DHCPDISCOVER messages
  - Extracts MAC addr and, if not in table, adds MAC addr and hostname to table
- For every new entry:
  - Rebuilds /etc/hosts and /etc/dhcpd.conf
  - Reconfigures NIS
  - Restarts DHCP and PBS
- Hostname is
  - <basename>-<cabinet>-<chassis>
- Configurable to change hostname
  - E.g., when adding new cabinets



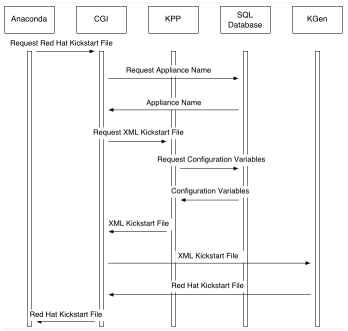
Configuration Derived from Database



Creating Kickstart File

- Node makes HTTP request to get configuration
  - Can be online or captured to a file
  - Node reports architecture type, IP address, [*appliance type*], [*options*]
- Kpp – preprocessor
  - Start at appliance type (node) and make a single large XML file by traversing the graph
  - Node-specific configuration looked up in SQL database
- Kgen – generation
  - Translation to kickstart format
  - Other formats could be supported

Generating Kickstart Files



Rocks Basic High Level Steps

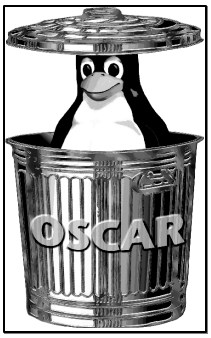
1. Install head node - *Boot Rocks-augmented CD*
2. Configure Cluster Services on Head Node – *in step 1*
3. Define Configuration of a Compute Node – *basic setup installed, can edit graph or nodes to customize*
4. For each compute node – repeat
   a) Detect Ethernet hardware address of new node – *use insert-ethers tool*
   b) Install complete OS on new node - *Kickstart*
   c) Complete configuration of new node – *in Kickstart*
5. Restart services on head node that are cluster aware (e.g. PBS, Sun Grid Engine) – *part of insert-ethers*

OSCAR

Open Source Cluster Application Resources

Installed and configured items:

- Head node services, e.g. DHCP, NFS
- Internal cluster networking configured
- SIS bootstraps compute -node installation, OS installed via network (PXE) or floppy boot
- OpenSSH/OpenSSL configured
- C3 power tools setup
- OpenPBS and MAUI installed and configured
- Install message passing libs: LAM/MPI, MPICH, PVM
- Env-Switcher/Modules installed and defaults setup



OSCAR Components

- Functional Areas
  - Cluster Installation
  - Programming Environment
  - Workload Management
  - Security
  - General Administration & Maintenance
- Other
  - Packaging
  - Documentation

Cluster Installation

- Linux Utility for cluster Install (LUI)
  - Builds cluster nodes from ground up
  - Maintains cluster information database
  - Uses RPM standard, simplifying software installation and maintenance
  - Heterogeneous Nature – Resource Based

Programming Environment

- Message Passing Paradigm
  - PVM – Parallel Virtual Machine
  - MPI – Message Passing Interface
    - MPICH
    - LAM/MPI

Workload Management

- Portable Batch System (OpenPBS)
  - Job management
  - Resource management
  - Default FIFO scheduler
- Maui Scheduler

Security

- OpenSSL
  - Open source implementation of the Secure Sockets Layer (SSL) protocol providing secure communications over a network
  - Export restricted
- OpenSSH
  - Open source implementation of the Secure Shell (SecSH) providing secure login, file transfer, and connections forwarding
  - Requires external encryption libraries → OpenSSL

General Administration & Maintenance

- Cluster Command & Control (C3)
  - Efficiently manage clusters where each node contains its own copy of OS & software
  - Functionality: cluster-wide command execution, file distribution & gathering, remote shutdown & restart, process status & termination, system image updates

OSCAR 1.3 – *base pkgs*

Package Name	Version
SIS	0.90-1/2.1.3oscar-1/1.25-1
C3	3.1
OpenPBS	2.2p11
MAUI	3.0.6p9
LAM/MPI	6.5.6
MPICH	1.2.4
PVM	3.4.4+6
Ganglia	2.2.3
Env-switcher/modules	1.0.4/3.1.6

OSCAR Cluster Installation Overview

- Set up hardware
- Install Linux on server (RedHat 7.1)
- Get OSCAR distribution & unpack
- Do cluster install
  - OSCAR Wizard guides users through the seven step process
- Test the cluster

OSCAR cluster view

- Server node – (1)
  - Service client requests
  - Gateway to external network
  - User home directories (NFS mounted)
  - Runs PBS server and scheduler
- Client nodes – (many)
  - Dedicated to computation
  - On private network
  - Local copy of OS

Hardware Considerations

- Server & Clients
  - Must be x86
  - Must be connected by an Ethernet network (preferably a private one)
- Clients
  - Must contain identical hardware
  - PXE Enabled NIC or Floppy Drive

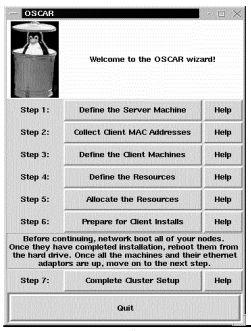
Install Linux on Server

- Distribution used must support RPM
- Needs to have X
- Can use machine with Linux already installed as server, otherwise a typical workstation install is sufficient

Installation Procedure

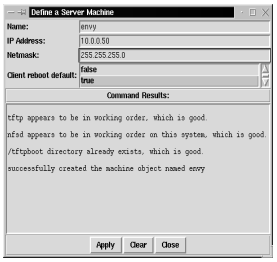
- Install RedHat on head node (*see also next slide*)
  - Include X Window support
  - Configure external/internal networking (*eth0*, *eth1*)
  - Create RPM directory, and copy RPMs from CD(s)
- Download OSCAR
  - Available at: http://oscar.sourceforge.net/
  - Extract the tarball (*see also next slide*)
- See installation guide (docs/ oscar_installation)
  - pdf and postscript versions available
- Run wizard (install_cluster eth*X*) to begin the install
- ./install_cluster *ethx*
  - *Ethx* is the Ethernet interface to the cluster
  - Prepares server for OSCAR
    - Initializes environment (directories,files)
    - Installs necessary software
      - LUI, NFS, DHCP, TFTP, Syslinux, Etherboot
  - Starts OSCAR Wizard

Installation Procedure



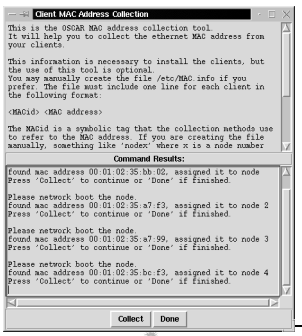
- Use OSCAR Wizard by completing the seven steps in sequence
- Each step has a quick help box
- Output displayed for each step

OSCAR Wizard – Step 1



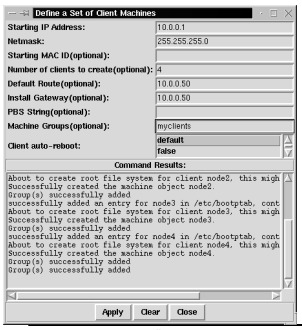
- Define the LUI server machine
  - Creates LUI server machine object
  - Enter name
    - Placeholder – does not have to be server's real host name
  - Enter IP & cluster subnet mask
  - Select client reboot default
    - Sets state for after LUI install process
      - Auto reboot or go to prompt
    - Set to false if –
      - BIOS settings must be changed
      - Using floppy to boot

OSCAR Wizard – Step 2



- Collect the client MAC addresses
  - The hard way – by hand...
    - Get from NIC sticker on board
    - Network boot and watch for MAC address
    - Enter in /etc/MAC.info
  - A little easier – use the collection utility
    - Sequential process
  - Much easier – advance planning required…
    - Vendor supplied file
    - Copy to /etc/MAC.info

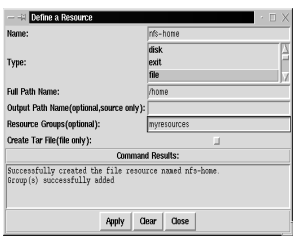
OSCAR Wizard – Step 3



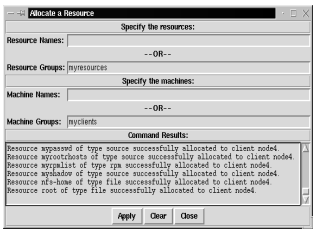
- Update /etc/hosts with client info
- Define clients to LUI
  - starting IP, cluster netmask, # of clients, client group name
  - default route & gateway
    - Not configured for IP forwarding or routing from internal to external net
  - PBS string
    - Used for resource management and scheduling
    - Not needed since all nodes the same
  - auto-reboot
    - Default – same setting for server from step 1
    - True = auto reboot
    - False = prompt
  - Machine Groups – USE THEM or regret it in step 5…

OSCAR Wizard – Step 4

- Define client resources
  - disk table – how to partition
  - file systems – one for each disk table
  - initial ramdisk – to support SCSI & NICS if no /etc/modules.conf source resource
  - RPM list – OSCAR default supplied
  - Optional – do for custom kernel only
    - Kernel
    - System map
  - Resource Groups – USE THEM or regret it in step 5…

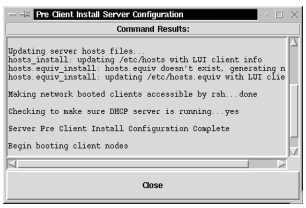


OSCAR Wizard – Step 5



- Allocate the resources to the LUI clients
  - Step 4 resources
  - Step 3 clients
- Easy, if Groups used in steps 3 & 4 – otherwise enter each…

OSCAR Wizard – Step 6



- Pre Client Install Server Configuration
  - OSCAR resources
    - Defines and allocates clients resources as required by OSCAR
  - C3
    - Tools and man pages installed
  - DHCP
    - Generates /etc/dhcpd.conf from MAC and IP information gathered earlier
  - hosts files
    - Generates entries in /etc/hosts and /etc/hosts.equiv for client machines

Client Installations

- Network Boot Clients
  - Preboot eXecution Environment (PXE)
    - Use PXE v2.0 or later – more stable
    - BIOS boot option setup may be required
      - Slow manual process for each box – yuck!
    - Not supported by all BIOSes & NICs
      - Some NIC & BIOS combos may try to fool you…
  - Etherboot
    - Floppy based
    - Typically used for older NICs

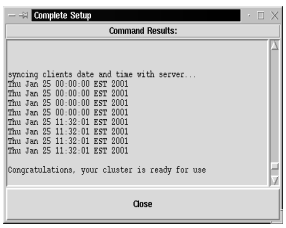
Client Installations

- Client downloads kernel from server
- Runs clone script as last startup item
  - Partitions disk & creates file systems
  - Mounts new file systems on /mnt
  - Chroots to /mnt & installs RPMs
  - Copies any source resources
  - Unmounts /mnt
- Goes to prompt, or reboots – as you specified

Client Installations

- A log of each client's progress is kept on the server
  - In /tftpboot/lim/log/<nodename>.log
- Continue with step 7 of wizard, when all the clients have finish ed their install and have been rebooted from their local disks
  - If changed in previous stage, reset BIOS boot option

OSCAR Wizard – Step 7



- Post Client Install Cluster Configuration
  - OpenSSL/SSH
    - Configured to allow access w/o password to client nodes for all users on server
  - SystemImager
  - PVM
  - MPICH, LAM/MPI
    - Creates machines file
  - OpenPBS
    - Configure server
    - Create server's nodes file
    - Configures xpbs admin gui & xpbsmon batch system monitor
  - Maui

Testing the Cluster

- OSCAR Cluster Test
  - PBS - name & date script
  - MPI - calculate pi (cpi)
  - PVM - master-slave
  - MPI & PVM tests run under PBS
- OSCAR benchmark suite removed as of v1.0 due to license issues

OSCAR Basic High Level Steps

1. Install head node - *Hand install using Distribution installer*
2. Configure Cluster Services on Head Node – *Follow installer setup script*
3. Define Configuration of a Compute Node – *Use OSCAR wizard to define a client image*
4. For each compute node – repeat
   a) Detect Ethernet hardware address of new node – *use OSCAR wizard*
   b) Install complete OS on new node – *SIS disk image downloaded and installed*
   c) Complete configuration of new node – *most customization already done in image*
5. Restart services on head node that are cluster aware (e.g. PBS, Sun Grid Engine) – *part of OSCAR install wizard*

Credits for Slides Used

- ROCKS
  - Mason Katz
  - Greg Bruno
  - Philip Papadopoulos
  - San Diego Supercomputer Center
- OSCAR
  - Stephen Scott
  - Thomas Naughton
  - Oak Ridge National Laboratory

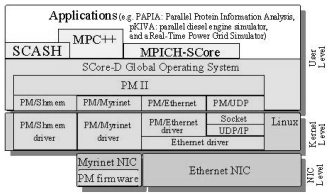
Other Toolkits

- Score
- LCFG
- XCat
- Chiba City Toolkit

SCore

- Single System Image
- Multiple Network Support
- Seamless Programming Environment
- Heterogeneous Programming Language
- Multiple Programming Paradigms
- Parallel Programming Support
- Fault Tolerance
- Flexible Job Scheduling

- http://cluster.mcs.st-and.ac.uk/score/en/



LCFG

- LCFG is a system for automatically installing and managing the configuration of large numbers of Unix systems. It is particularly suitable for sites with very diverse and rapidly changing configurations.
- Description based
- Proprietary configuration language, custom compiler to create XML, uses own boot environment

XCat

- xCAT (Extreme Cluster Administration Toolkit)
- Limited support for SuSE Linux YaST
- License limited to IBM hardware
- A lot of initial description and scripting necessary
- Integrated with IBM poprietary management processor
  - BIOS updates, remote power cycling, etc

Chiba City Toolkit

- Unsupported collection of tools from ANL
- Image based installer
- See Chapter 20 for longer description