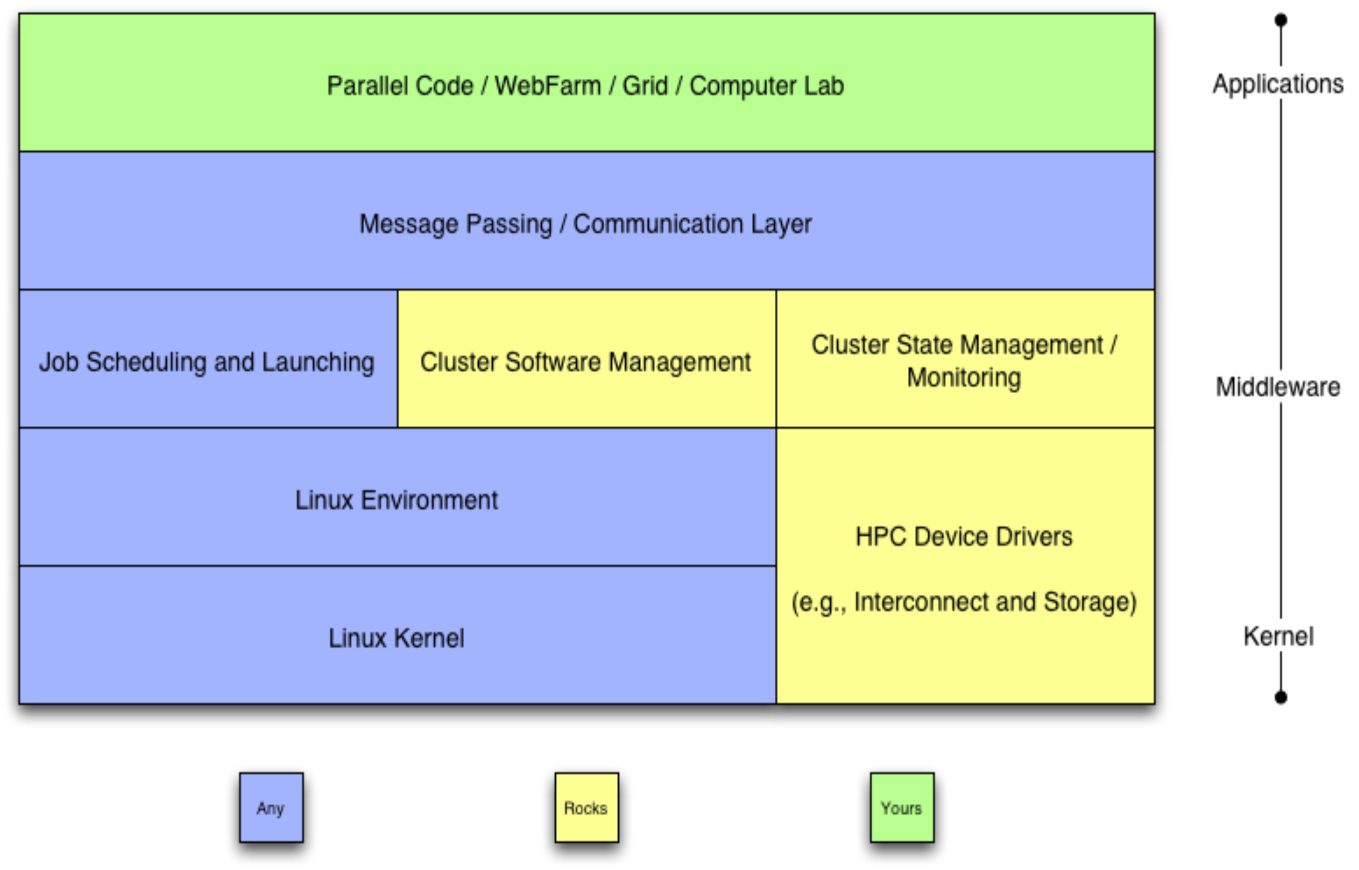
Introduction to Rocks

Rocks-A-Palooza II

Overview of Rocks

The Rocks software stack

Cluster Software Stack



Parallel Code / WebFarm / Grid / Computer Lab

Message Passing / Communication Layer

Job Scheduling and Launching

Cluster Software Management

Cluster State Management / Monitoring

Linux Environment

Linux Kernel

HPC Device Drivers

(e.g., Interconnect and Storage)

Applications

Middleware

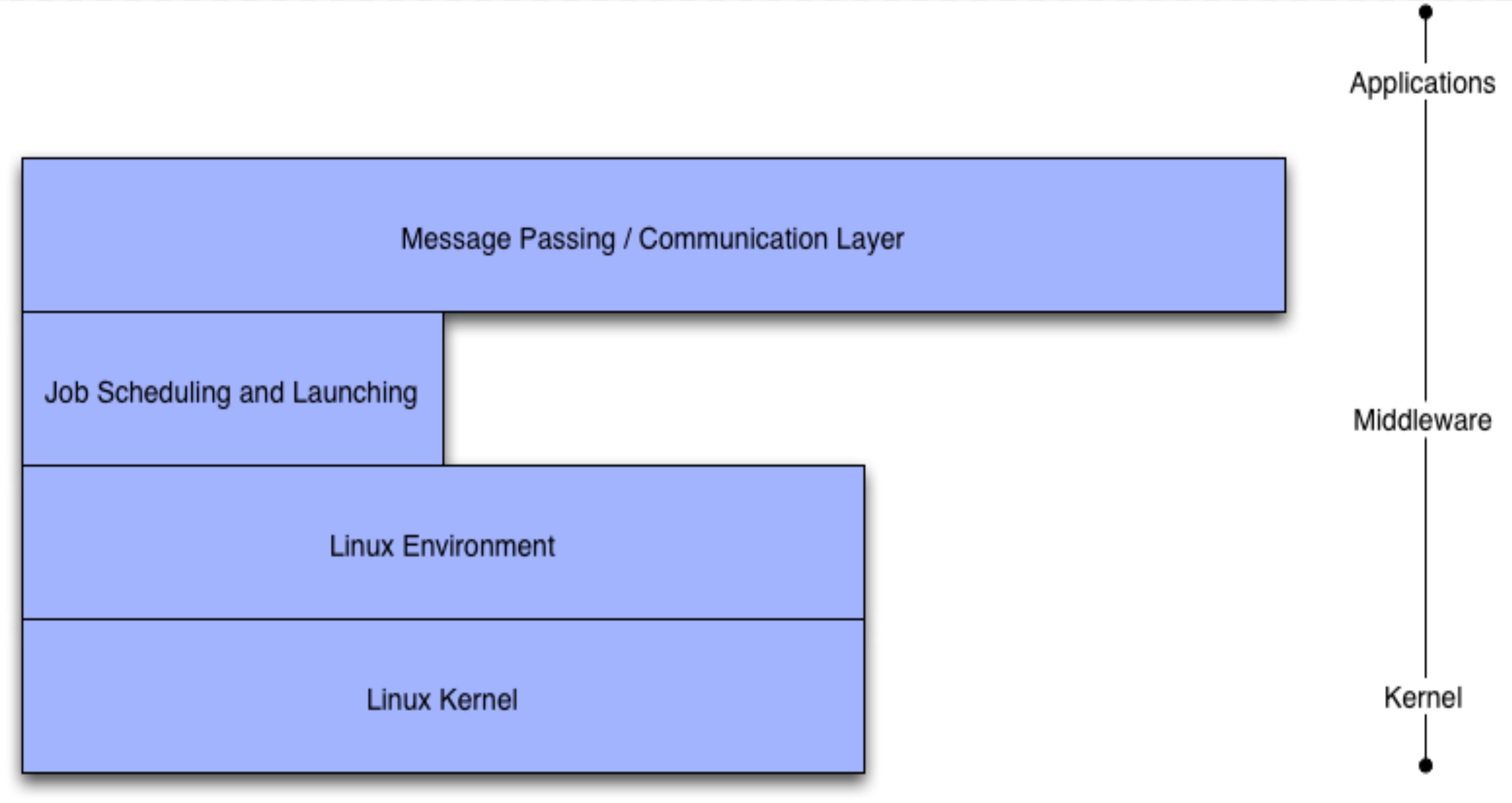
Kernel

Any

Rocks

Yours

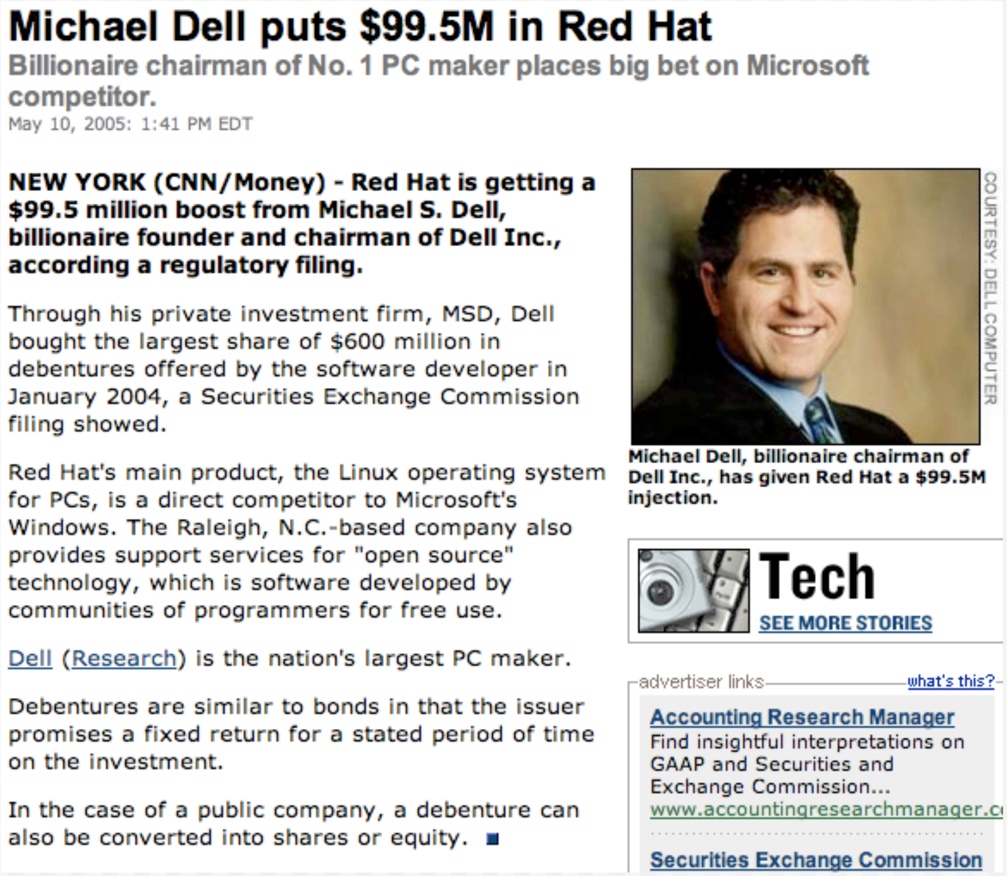
Common to Any Cluster


Applications
Message Passing / Communication Layer
Job Scheduling and Launching
Middleware
Linux Environment
Linux Kernel
Kernel

Red Hat

- Enterprise Linux 4.0
  - Recompiled from public SRPMS, including errata updates (source code)
  - No license fee required, redistribution is also fine
  - Recompiled for all CPU types (x86, Opteron, Itanium)
  - *Rocks 5.0 will be based on RHEL 5.0 (Centos, or RHEL)*
- Standard Red Hat Linux kernel
  - No Rocks added kernel patches
- No support for other distributions
  - Red Hat is the market leader for Linux
    - In the US
    - And becoming so in Europe
  - Trivial to support any Anaconda-based system
  - Others would be harder, and require vendor support (SuSe ~ 12 months work)
- Excellent support for automated installation
  - Scriptable installation (Kickstart)
  - Very good hardware detection

Dell Invests in Red Hat



Michael Dell puts $99.5M in Red Hat

Billionaire chairman of No. 1 PC maker places big bet on Microsoft competitor.

May 10, 2005: 1:41 PM EDT

NEW YORK (CNN/Money) - Red Hat is getting a $99.5 million boost from Michael S. Dell, billionaire founder and chairman of Dell Inc., according a regulatory filing.

Through his private investment firm, MSD, Dell bought the largest share of $600 million in debentures offered by the software developer in January 2004, a Securities Exchange Commission filing showed.

Red Hat's main product, the Linux operating system for PCs, is a direct competitor to Microsoft's Windows. The Raleigh, N.C.-based company also provides support services for "open source" technology, which is software developed by communities of programmers for free use.

Dell (Research) is the nation's largest PC maker.

Debentures are similar to bonds in that the issuer promises a fixed return for a stated period of time on the investment.

In the case of a public company, a debenture can also be converted into shares or equity. ■

COURTESY: DELL COMPUTER

Michael Dell, billionaire chairman of Dell Inc., has given Red Hat a $99.5M injection.

Batch Systems

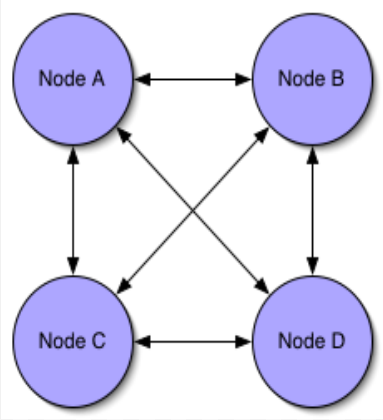
- Portable Batch System and Maui
  - Long time standard for HPC queuing systems
  - Maui provides backfilling for high throughput
  - PBS/Maui system can be fragile and unstable
  - Multiple code bases:
    - PBS
    - OpenPBS
    - PBSPro
    - Scalable PBS
- Sun Grid Engine
  - Rapidly becoming the new standard
  - Integrated into Rocks by Scalable Systems
    - See Najib
  - Now the default scheduler for Rocks
  - Robust and dynamic

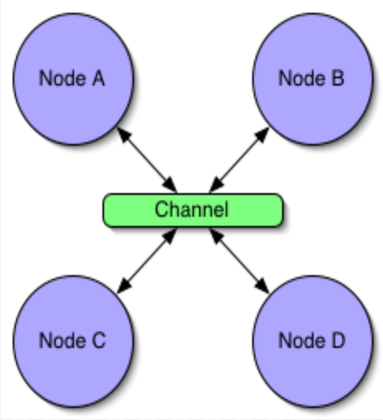
Communication Layer

- None
  - “Embarrassingly Parallel”
- Sockets
  - Client-Server model
  - Point-to-point communication
- MPI - Message Passing Interface
  - Message Passing
  - Static model of participants
- PVM - Parallel Virtual Machines
  - Message Passing
  - For Heterogeneous architectures
  - Resource Control and Fault Tolerance

Sockets are low level

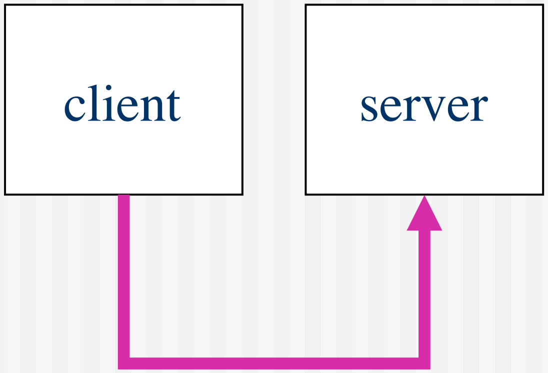
- Sockets
  - Point-to-Point
  - N machines = (n^2 - n)/2 connections
  - 1, 3, 6, 10, 15, …
- MPI/PVM
  - Shared virtual channel
  - Implementation could be sockets
  - Easier to program



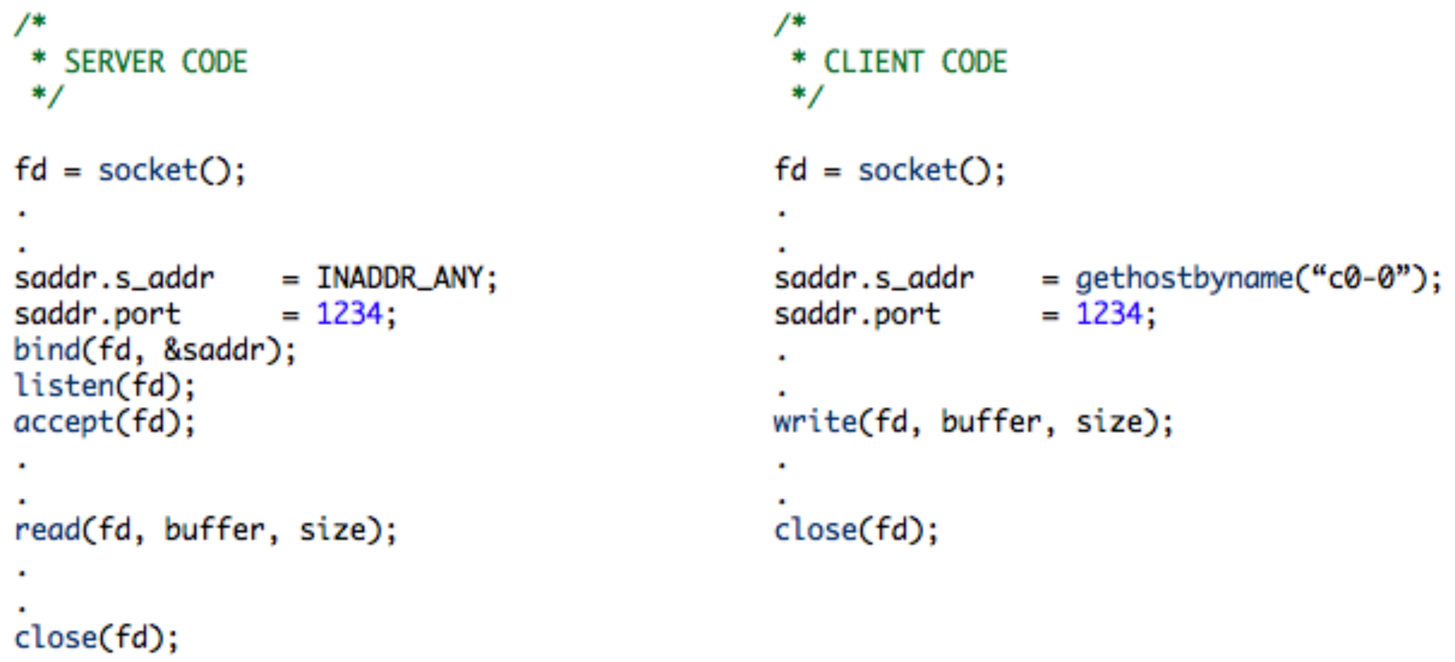


Sockets

- Open an endpoint
- Specify IP address and port
- Send / receive messages
  - If TCP, only point-to-point messages
  - If UDP, option of point-to-point or multicast (broadcast)
- Shutdown connection



High-level TCP Example



```
/*
 * SERVER CODE
 */

fd = socket();
.
.
saddr.s_addr     = INADDR_ANY;
saddr.port       = 1234;
bind(fd, &saddr);
listen(fd);
accept(fd);
.
.
read(fd, buffer, size);
.
.
close(fd);
```

```
/*
 * CLIENT CODE
 */

fd = socket();
.
.
saddr.s_addr     = gethostbyname(“c0-0”);
saddr.port       = 1234;
.
.
write(fd, buffer, size);
.
.
close(fd);
```

Challenges with Sockets

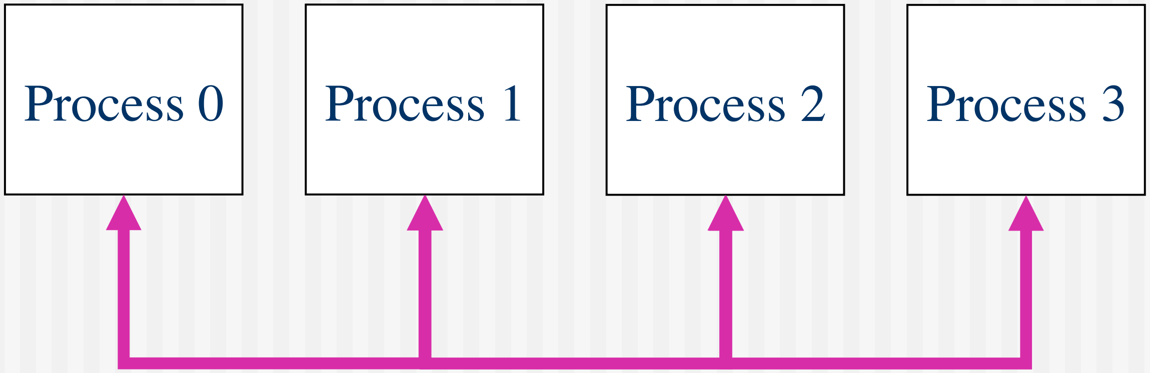
- TCP
  - Reliable, but byte oriented
  - Need to write code to send and receive *packets* (at the application level)
- UDP
  - Unreliable
  - Need to write code to reliably send packets

MPI

- Message Passing Interface
- De facto standard for message passing
  - Runs over many CPU architectures and many communication substrates
- There are (and were) lots of good messaging libraries
  - But, MPI is the most pervasive
  - Developed a practical, portable, efficient and flexible standard
  - In development since 1992

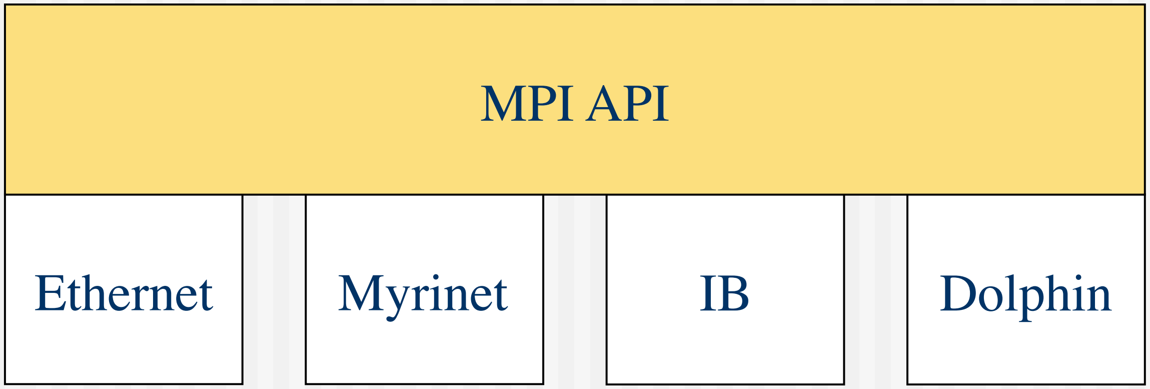
MPI

- Explicitly move data like sockets, but virtualizes the endpoints
  - Remote endpoints addressed by integer 0, 1, …, n
- Primitives to support point-to-point and broadcast



MPI

- Single interface to pass messages over many communication substrates



MPI API

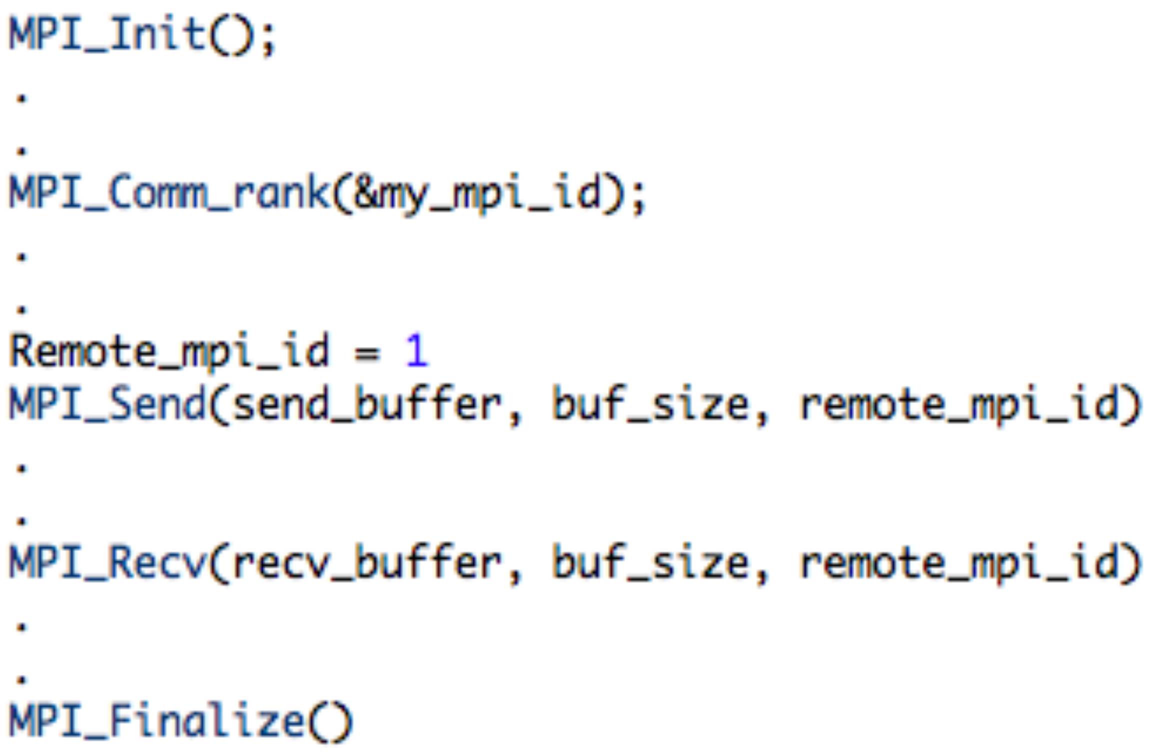
Ethernet

Myrinet

IB

Dolphin

High-level MPI Example



```
MPI_Init();
.
.
MPI_Comm_rank(&my_mpi_id);
.
.
Remote_mpi_id = 1
MPI_Send(send_buffer, buf_size, remote_mpi_id)
.
.
MPI_Recv(recv_buffer, buf_size, remote_mpi_id)
.
.
MPI_Finalize()
```

Challenges with MPI

- If a node fails, no easy way to reconfigure and route around the problem
  - Basically, your program stops

Compile

- MPICH with GNU Compilers and Ethernet

Compiler	**Path**
`C:`	`/opt/mpich/ethernet/gcc/bin/mpicc`
`C++:`	`/opt/mpich/ethernet/gcc/bin/mpiCC`
`F77:`	`/opt/mpich/ethernet/gcc/bin/mpif77`

- MPICH with GNU Compilers and Myrinet

Compiler	**Path**
`C:`	`/opt/mpich/myrinet/gcc/bin/mpicc`
`C++:`	`/opt/mpich/myrinet/gcc/bin/mpiCC`
`F77:`	`/opt/mpich/myrinet/g77/bin/mpif77`

Compile

- MPICH with Intel Compilers and Ethernet

Compiler	**Path**
C:	/opt/mpich/ethernet/ecc/mpicc
C++:	/opt/mpich/ethernet/ecc/mpiCC
F77:	/opt/mpich/ethernet/ecc/mpif77
F90:	/opt/mpich/ethernet/ecc/mpif90

- MPICH with Intel Compilers and Myrinet

Compiler	**Path**
C:	/opt/mpich/myrinet/ecc/mpicc
C++:	/opt/mpich/myrinet/ecc/mpiCC
F77:	/opt/mpich/myrinet/efc/mpif77
F90:	/opt/mpich/myrinet/efc/mpif90

PVM

- Parallel Virtual Machines v3.4.3
  - Message passing interface for heterogeneous architectures
    - Supports over 60 variants of UNIX
    - Supports Windows NT
  - Resource control and meta computing
  - Fault tolerance
  - http://www.csm.ornl.gov/pvm/

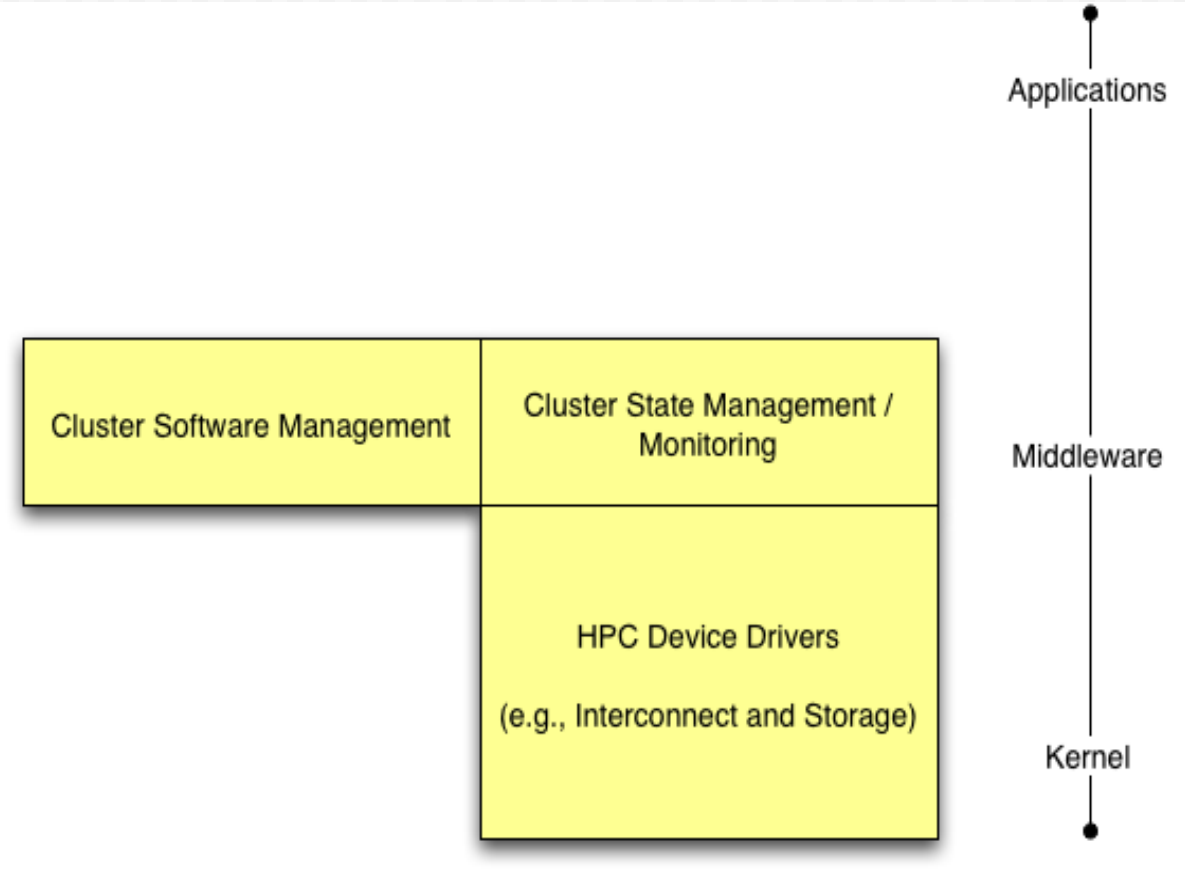
NFS

- User account are served over NFS
  - Works for small clusters (<= 128 nodes)
  - Will not work for large clusters (>1024 nodes)
  - NAS is better than Linux
    - Rocks uses the Frontend machine to server NFS
    - We have deployed NAS on several clusters
- Applications are not served over NFS
  - /usr/local/ does not exist
  - All software is installed locally from RPM

Open SSH

- Replaces Telnet, Rsh
  - Cryptographically strong authentication and encryption
  - Forwards X11 connections (no more $DISPLAY)
- Rocks uses SSH
  - Mpirun
  - Cluster-fork
- Ssh-agent
  - Manager for SSH keys
  - ssh-agent $SHELL

Rocks Cluster Software


Applications
Cluster Software Management
Cluster State Management / Monitoring
Middleware
HPC Device Drivers
(e.g., Interconnect and Storage)
Kernel

SNMP

- Enabled on all compute nodes
- Great for point-to-point use
  - Good for high detail on a single end-point
  - Does not scale to full cluster wide use
- Supports Linux MIB
  - Uptime, Load, Network statistics
  - Install Software
  - Running Processes

Syslog

- Native UNIX system event logger
  - Logs events to local dist
    - /var/log/message
    - Rotates logs daily, eventually historic data is lost
  - Forwards all message to the frontend
- Scalable
  - Can add additional loghosts
  - Can throttle verbosity of loggers
- Uses
  - Predicting hardware and software failures
  - Post Mortem on crashed nodes
  - Debugging System startup

eKV

- Remotely Interact with Installation
  - Initial kickstart
  - Re-Installation
- Shoot-node
  - Reinstall OS and brings up eKV
- eKV
  - Ssh to node while it is installing
  - See the console output over Ethernet

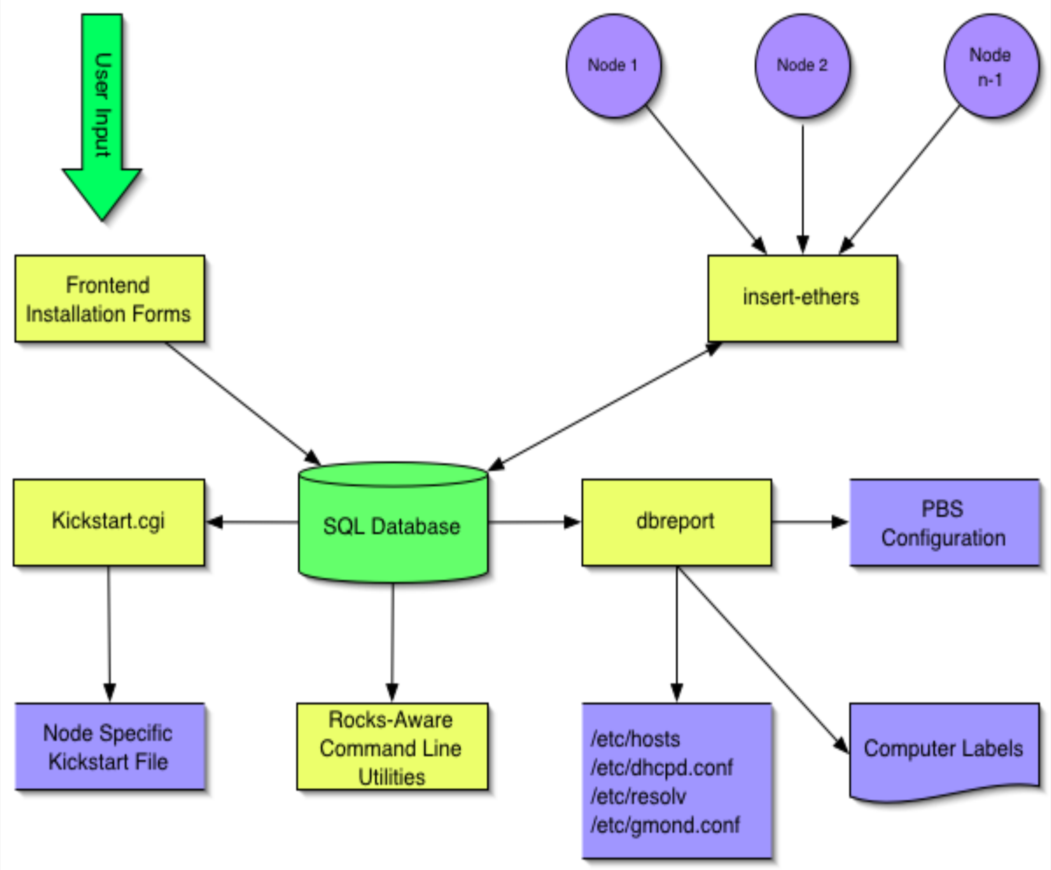
Cluster State Management

- Static Information
  - Node addresses
  - Node types
  - Site-specific configuration
- Dynamic Information
  - CPU utilization
  - Disk utilization
  - Which nodes are online

MySQL

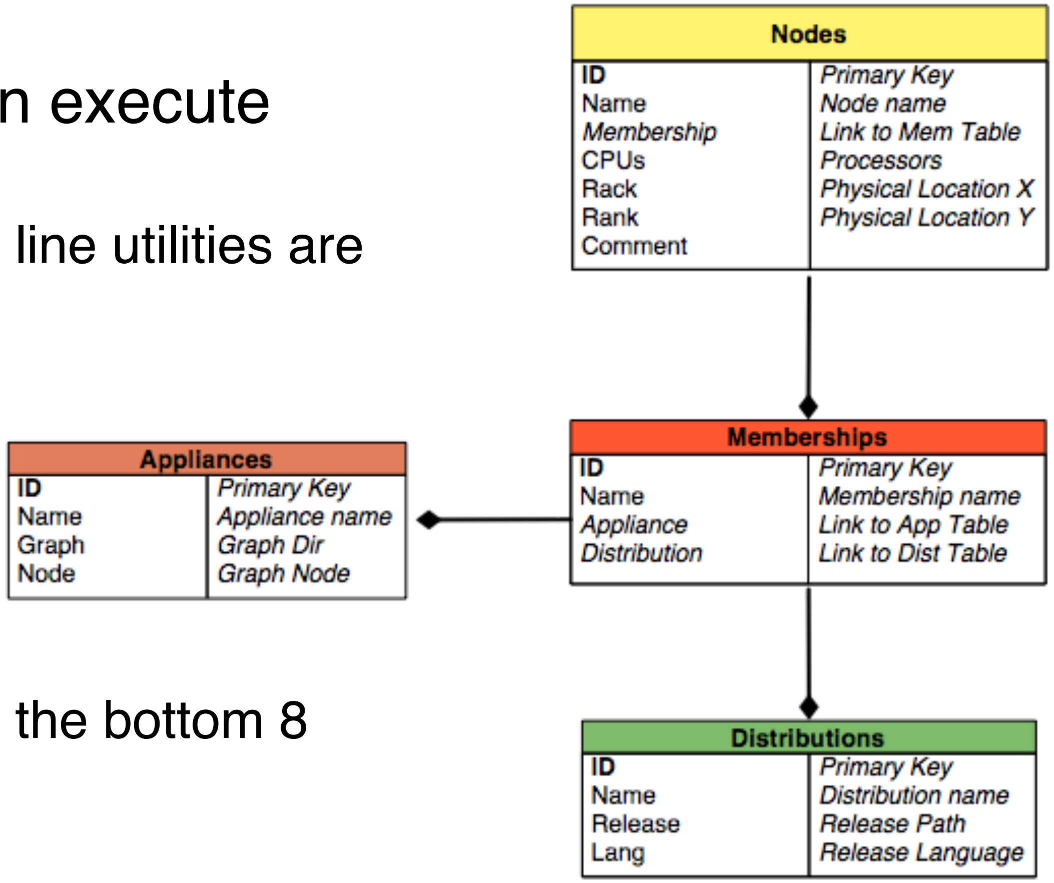
ganglia

Cluster Database



Node Info Stored In A MySQL Database

- If you know SQL, you can execute powerful commands
  - Rocks-supplied command line utilities are tied into the database
  - E.g., get the hostname for the bottom 8 nodes of each cabinet:



Nodes	
ID	*Primary Key*
Name	*Node name*
Membership	*Link to Mem Table*
CPUs	*Processors*
Rack	*Physical Location X*
Rank	*Physical Location Y*
Comment	

Memberships	
ID	*Primary Key*
Name	*Membership name*
Appliance	*Link to App Table*
Distribution	*Link to Dist Table*

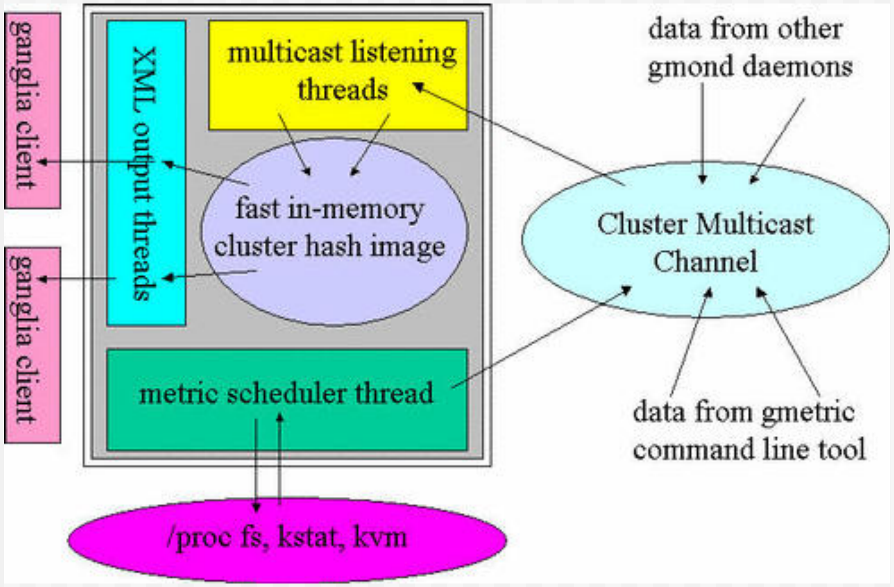
Appliances	
ID	*Primary Key*
Name	*Appliance name*
Graph	*Graph Dir*
Node	*Graph Node*

Distributions	
ID	*Primary Key*
Name	*Distribution name*
Release	*Release Path*
Lang	*Release Language*

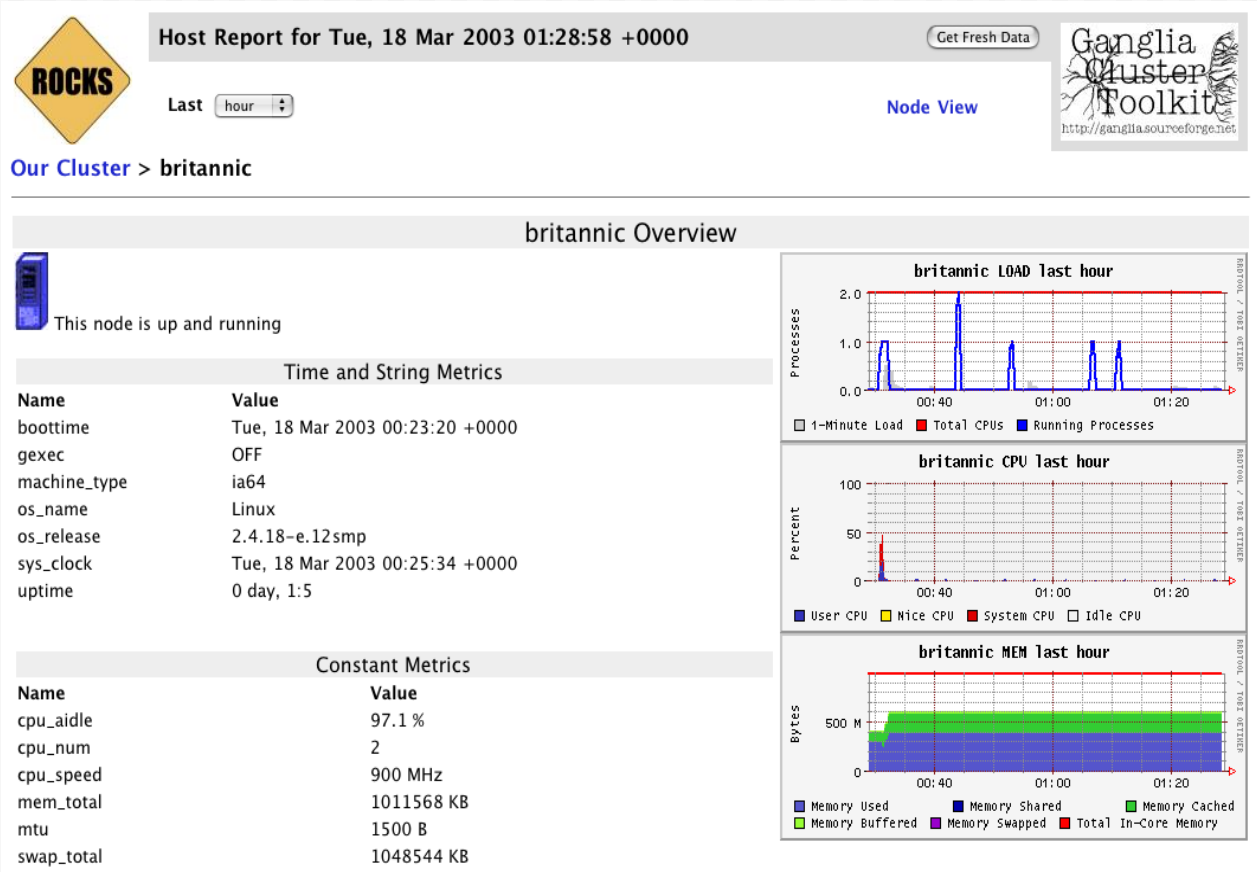
```
# cluster-fork --query="select name from nodes where rank<8" hostname
```

Ganglia (or SCMSWeb / SCE Roll)

- Scalable cluster monitoring system
  - Based on ip multi-cast
  - Matt Massie, et al from UCB
  - http://ganglia.sourceforge.net
- Gmon daemon on every node
  - Multicasts system state
  - Listens to other daemons
  - All data is represented in XML
- Ganglia command line
  - Python code to parse XML to English
- Gmetric
  - Extends Ganglia
  - Command line to multicast single metrics


ganglia client
ganglia client
XML output threads
multicast listening threads
fast in-memory cluster hash image
metric scheduler thread
/proc fs, kstat, kvm
data from other gmond daemons
Cluster Multicast Channel
data from gmetric command line tool

Ganglia Screenshot



Host Report for Tue, 18 Mar 2003 01:28:58 +0000

Our Cluster > britannic

britannic Overview

This node is up and running

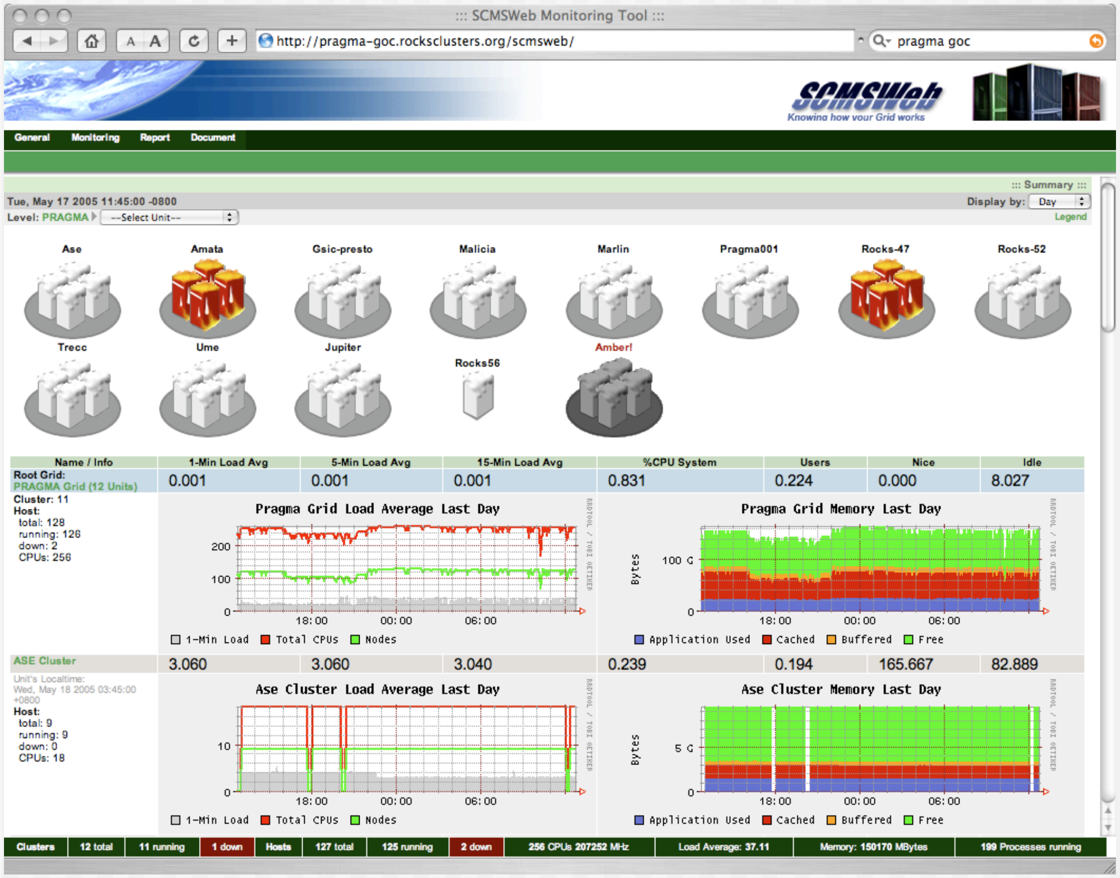
Time and String Metrics

Name	Value
boottime	Tue, 18 Mar 2003 00:23:20 +0000
gexec	OFF
machine_type	ia64
os_name	Linux
os_release	2.4.18-e.12smp
sys_clock	Tue, 18 Mar 2003 00:25:34 +0000
uptime	0 day, 1:5

Constant Metrics

Name	Value
cpu_aidle	97.1 %
cpu_num	2
cpu_speed	900 MHz
mem_total	1011568 KB
mtu	1500 B
swap_total	1048544 KB

SCMSWeb Screenshot



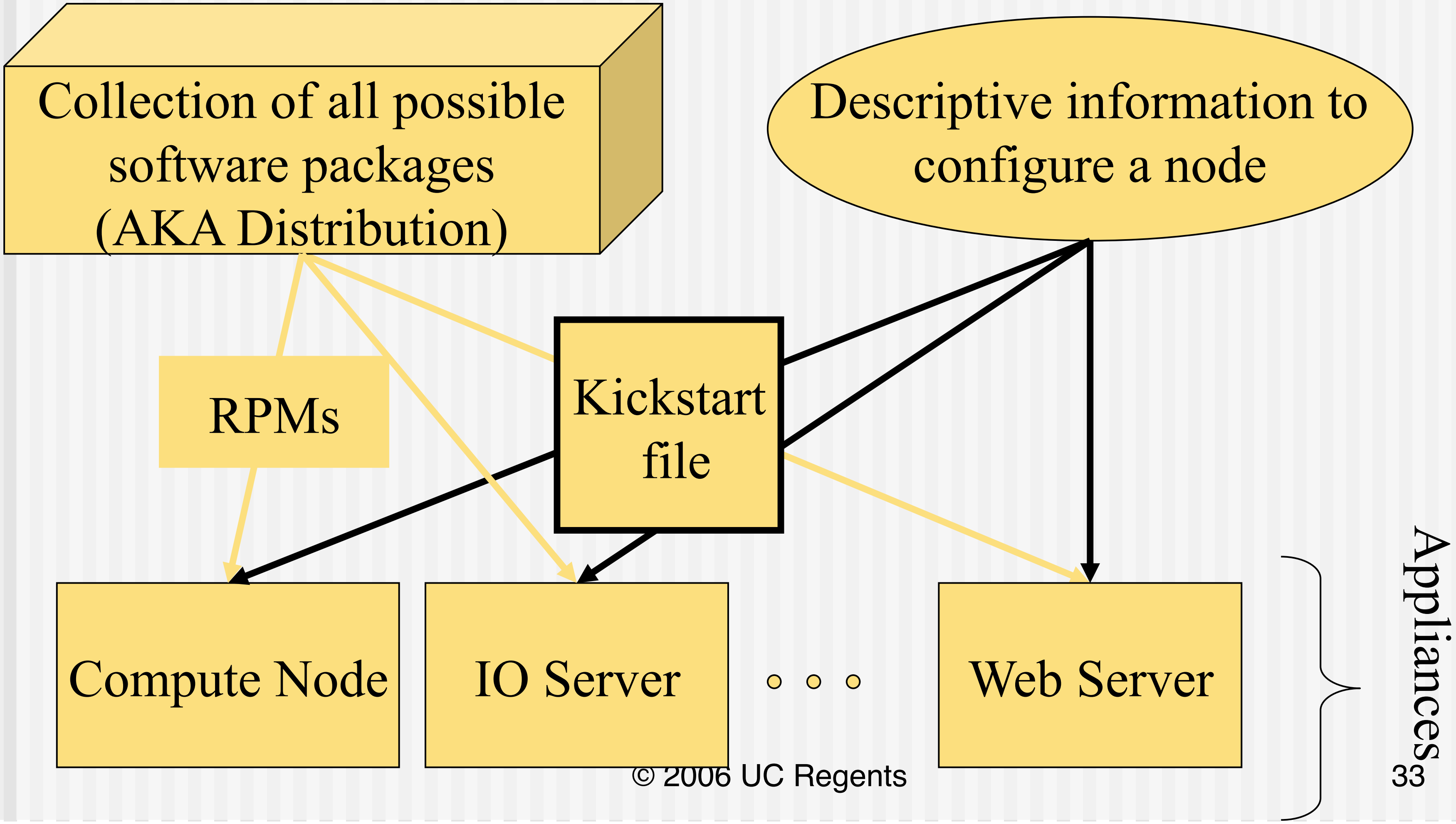
Tue, May 17 2005 11:45:00 -0800

Level: PRAGMA --Select Unit--

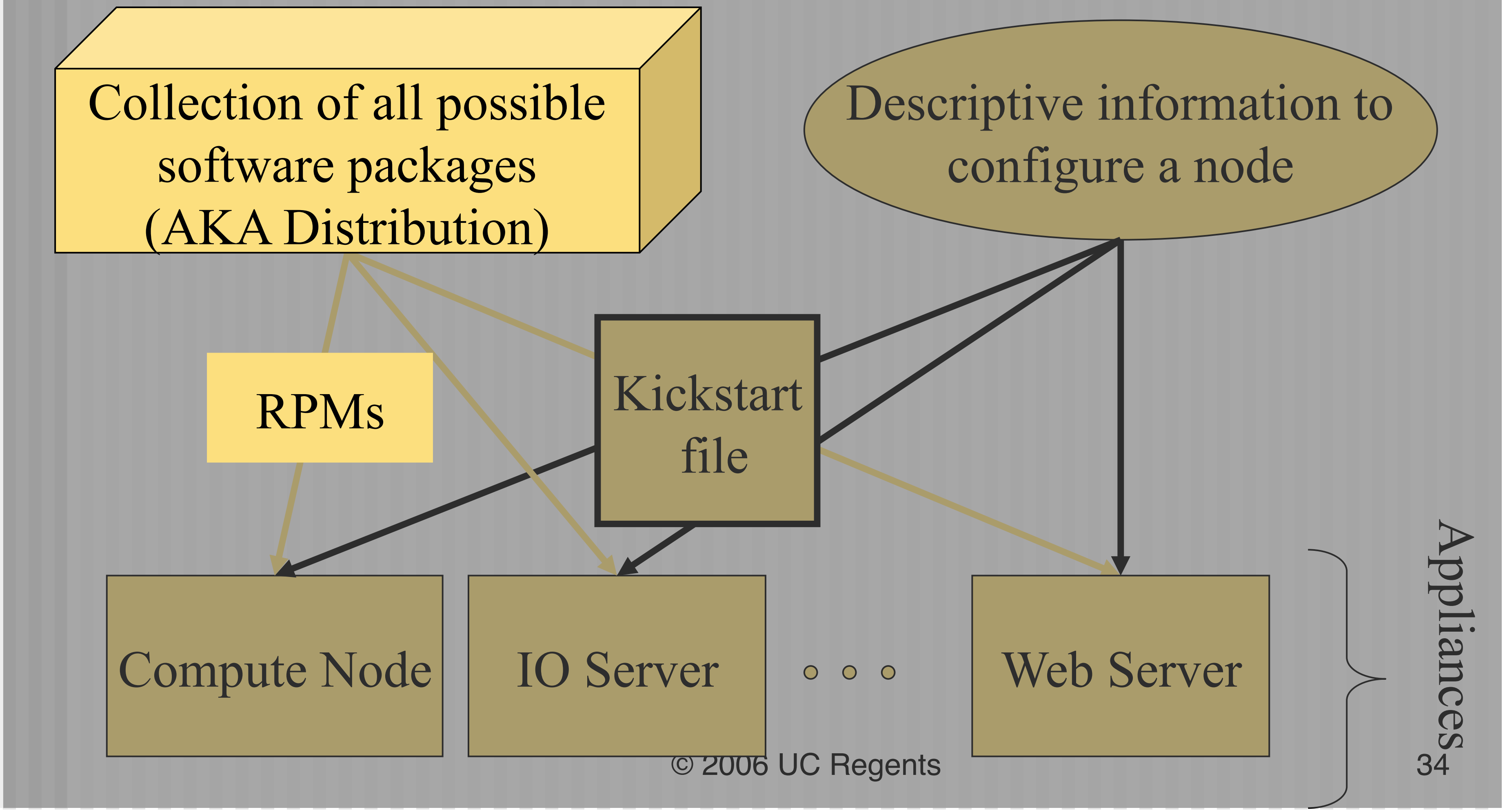
Name / Info	1-Min Load Avg	5-Min Load Avg	15-Min Load Avg	%CPU System	Users	Nice	Idle
Root Grid: PRAGMA Grid (12 Units)	0.001	0.001	0.001	0.831	0.224	0.000	8.027
Cluster: 11 Host: total: 128 running: 126 down: 2 CPUs: 256							
ASE Cluster	3.060	3.060	3.040	0.239	0.194	165.667	82.889
Unit's Localtime: Wed, May 18 2005 03:45:00 +0800 Host: total: 9 running: 9 down: 0 CPUs: 18							

Clusters	12 total	11 running	1 down	Hosts	127 total	125 running	2 down	256 CPUs 207252 MHz	Load Average: 37.11	Memory: 150170 MBytes	199 Processes running

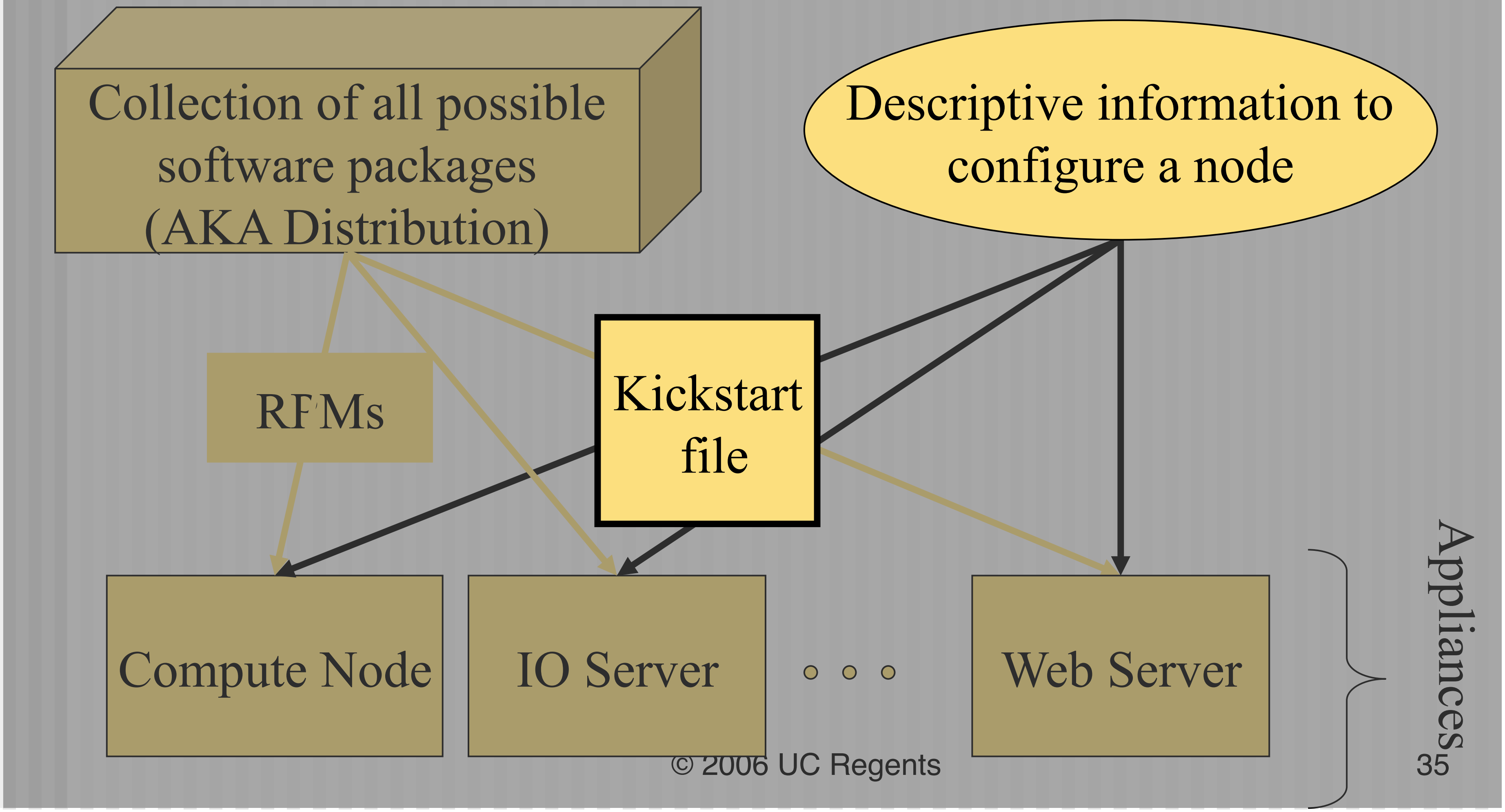
Software Installation


Collection of all possible software packages (AKA Distribution)
Descriptive information to configure a node
RPMs
Kickstart file
Compute Node
IO Server
○ ○ ○
Web Server
Appliances

Software Repository


Collection of all possible software packages (AKA Distribution)
Descriptive information to configure a node
RPMs
Kickstart file
Compute Node
IO Server
Web Server
Appliances

Installation Instructions


Collection of all possible software packages (AKA Distribution)
Descriptive information to configure a node
RPMs
Kickstart file
Compute Node
IO Server
○ ○ ○
Web Server
Appliances

Cluster Software Management

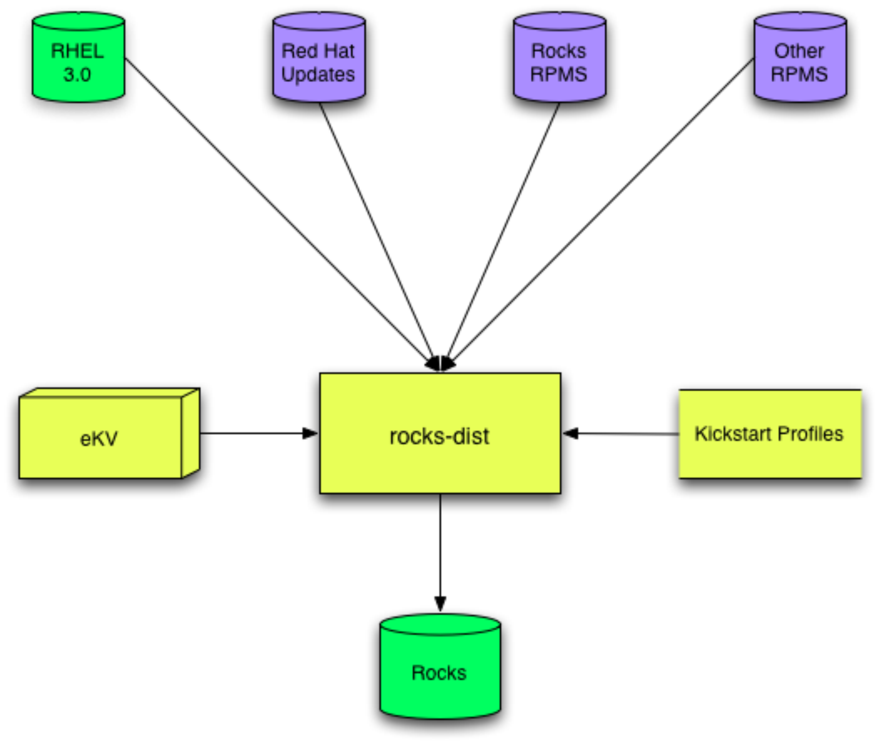
Software Packages

- RPMs
  - Standard Red Hat (desktop) packaged software
  - Or your own addons
- Rocks-dist
  - Manages the RPM repository
  - This is the distribution

Software Configuration

- Tuning RPMs
  - For clusters
  - For your site
  - Other customization
- XML Kickstart
  - Programmatic System Building
  - Scalable

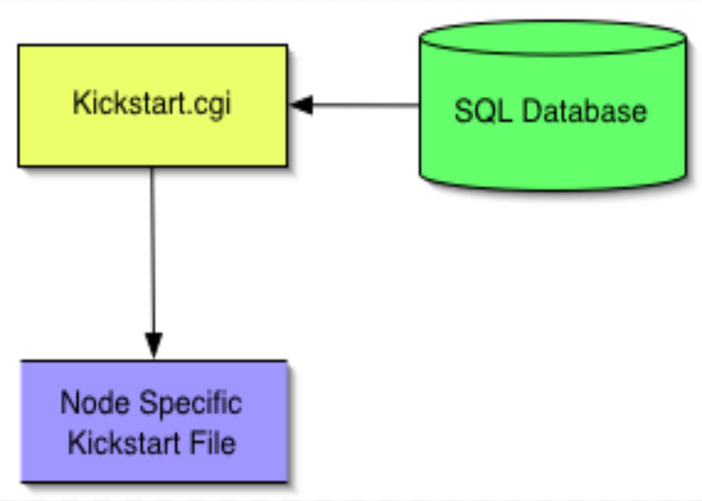
Building a Rocks Distribution



- Start with Red Hat
- Add updates, Rocks (and optional other) software
- Add Kickstart profiles
- Modify Red Hat installation boot image
- Resulting in a Red Hat compatible Rocks distribution

Kickstart

- Red Hat's Kickstart
  - Monolithic flat ASCII file
  - No macro language
  - Requires forking based on site information and node type.
- Rocks XML Kickstart
  - Decompose a kickstart file into nodes and a graph
    - Graph specifies OO framework
    - Each node specifies a service and its configuration
  - Macros and SQL for site configuration
  - Driven from web cgi script



Kickstart File Sections

- Main
  - Disk partitioning
  - Root password
  - RPM repository URL
  - …
- Packages
  - List of RPMs (w/o version numbers)
  - The repository determines the RPM versions
  - The kickstart file determines the set of RPMs
- Pre
  - Shell scripts run before RPMs are installed
  - Rarely used (Rocks uses it to enhance kickstart)
- Post
  - Shell scripts to cleanup RPM installation
  - Fixes bugs in packages
  - Adds local information

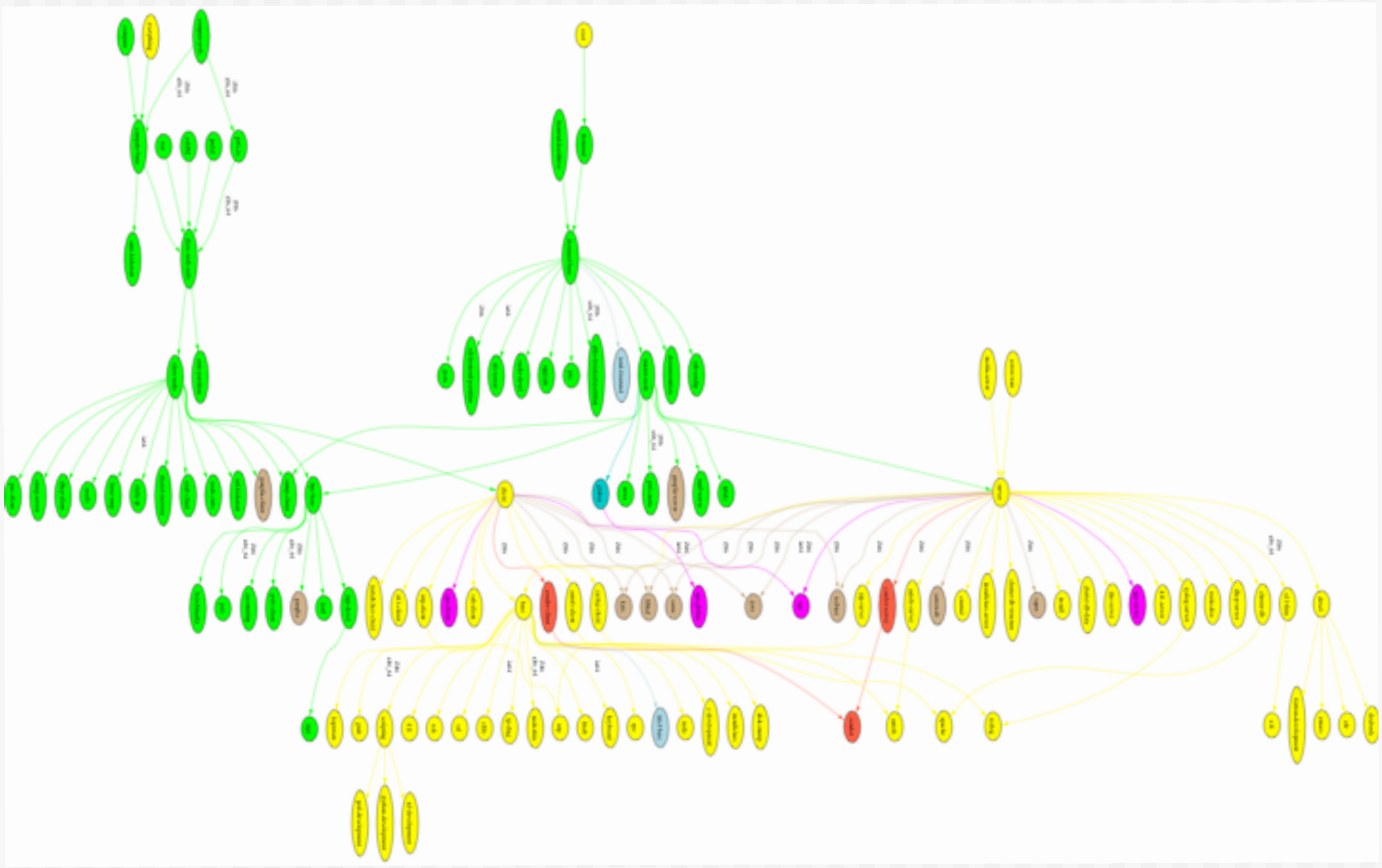
Sample Node File

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE kickstart SYSTEM "@KICKSTART_DTD@" [<!ENTITY ssh "openssh">]>
<kickstart>
            <description>
            Enable SSH
            </description>

            <package>&ssh;</package>
            <package>&ssh;-clients</package>
            <package>&ssh;-server</package>
            <package>&ssh;-askpass</package>
<post>

cat &gt; /etc/ssh/ssh_config &lt;&lt; 'EOF’ <!-- default client setup -->
Host *
        ForwardX11 yes
        ForwardAgent yes
EOF

chmod o+rx /root
mkdir /root/.ssh
chmod o+rx /root/.ssh

</post>
</kickstart>>
```

Sample Graph File

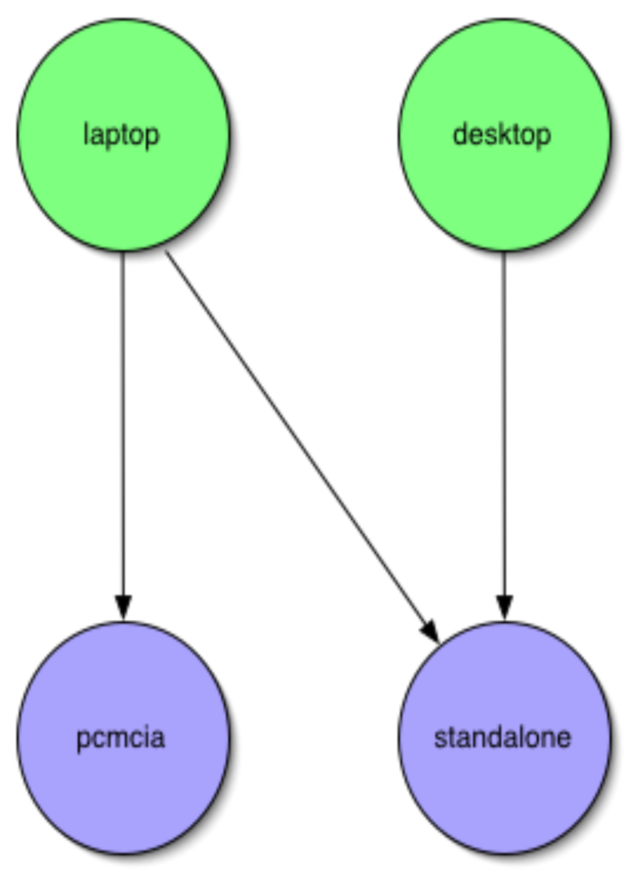
```
<?xml version="1.0" standalone="no"?>

<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=”grub"  arch="i386"/>
            <edge from="base" to="elilo" arch="ia64"/>
                        …
            <edge from="node" to="base"/>
            <edge from="node" to="accounting"/>
            <edge from="slave-node" to="node"/>
            <edge from="slave-node" to="nis-client"/>
            <edge from="slave-node" to="autofs-client"/>
            <edge from="slave-node" to="dhcp-client"/>
            <edge from="slave-node" to="snmp-server"/>
            <edge from="slave-node" to="node-certs"/>
            <edge from="compute" to="slave-node"/>
            <edge from="compute" to="usher-server"/>
            <edge from="master-node" to="node"/>
            <edge from="master-node" to="x11"/>
            <edge from="master-node" to="usher-client"/>
</graph>
```

Kickstart Framework



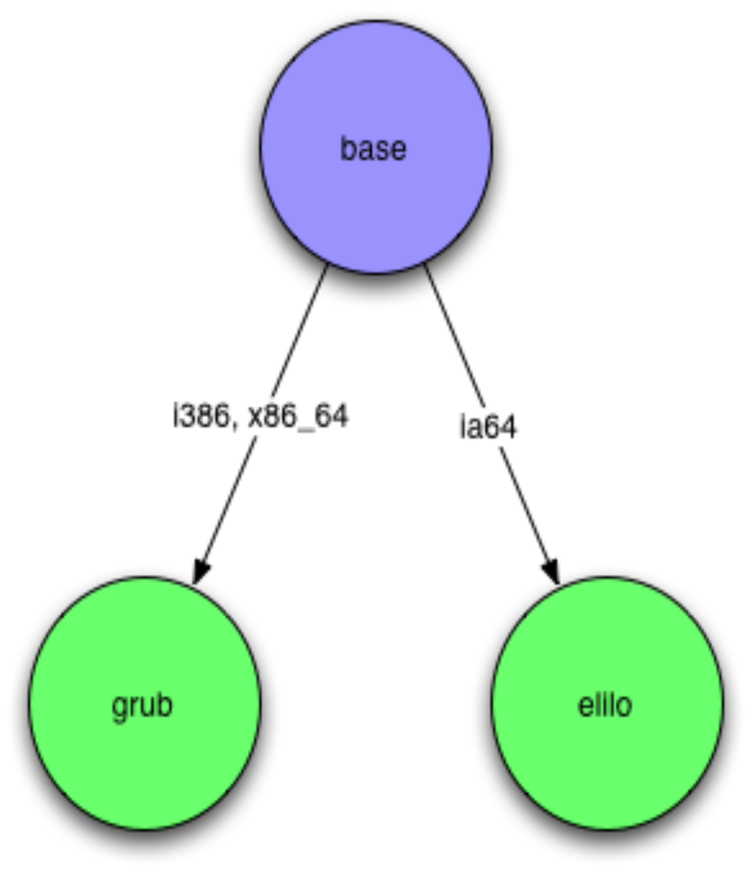
Appliances

- Laptop / Desktop
  - Appliances
  - Final classes
  - Node types
- Desktop IsA
  - standalone
- Laptop IsA
  - standalone
  - pcmcia
- Code re-use is good



Architecture Differences

- Conditional inheritance
- Annotate edges with target architectures
- if i386
  - Base IsA grub
- if ia64
  - Base IsA elilo
- One Graph, Many CPUs
  - Heterogeneity is easy
  - Not true for SSI or Imaging

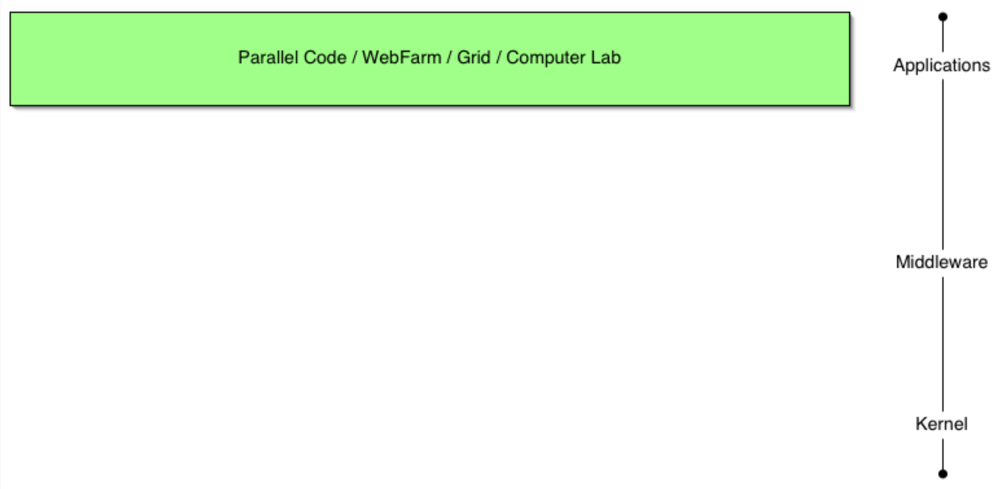


Optional Drivers

- PVFS
  - Parallel Virtual File System
  - Kernel module built for all nodes
  - User must decide to enable
- Myrinet
  - High Speed and Low Latency Interconnect
  - GM/MPI for user Applications
  - Kernel module built for all nodes with Myrinet cards
- Add your own
  - Cluster Gigabit Ethernet driver
  - Infiniband driver

Application Layer

- Rocks Rolls
  - Optional component
  - Created by SDSC
  - Created by others
- Example
  - Bio (BLAST)
  - Chem (GAMESS)
  - Visualization Clusters

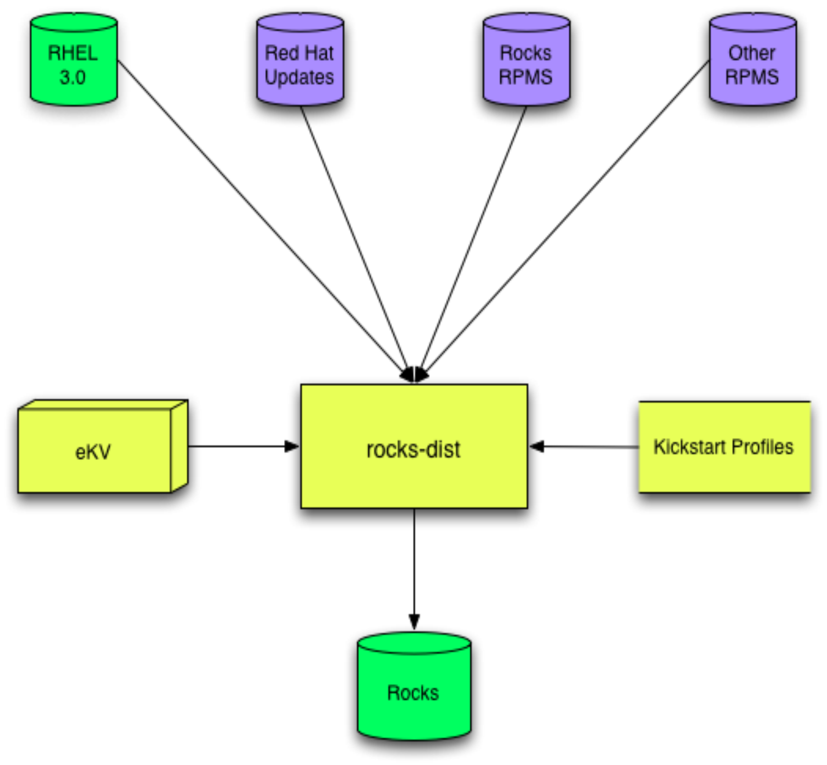


Building on Top of Rocks

Inheritance and Rolls

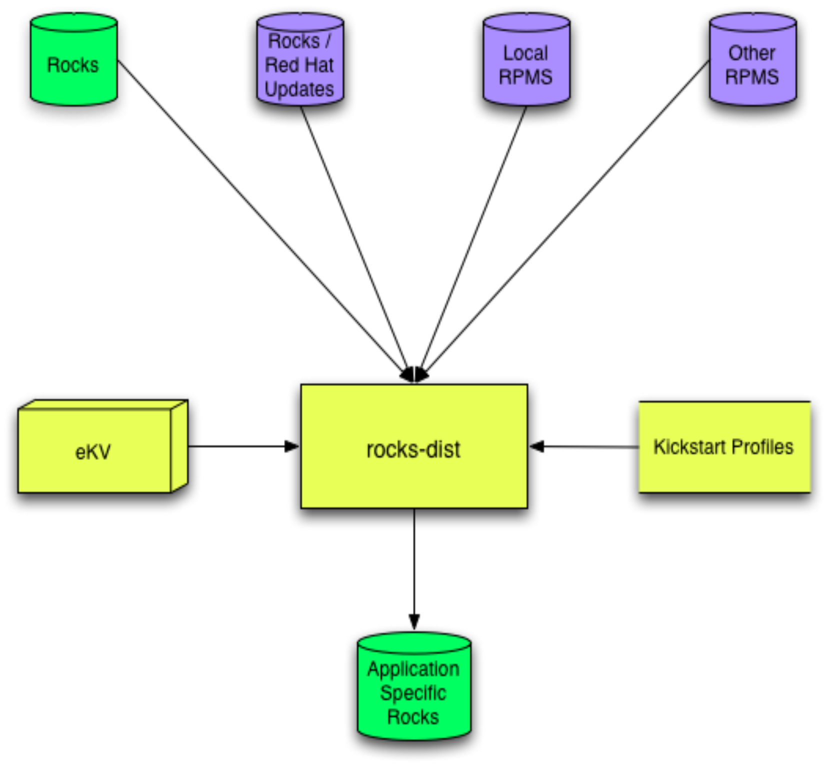
How Rocks in built

- Rocks-dist
  - Merges all RPMs
    - Red Hat
    - Rocks
  - Resolves versions
  - Creates Rocks
- Rocks distribution
  - Looks just like Red Hat
  - Cluster optimized Red Hat



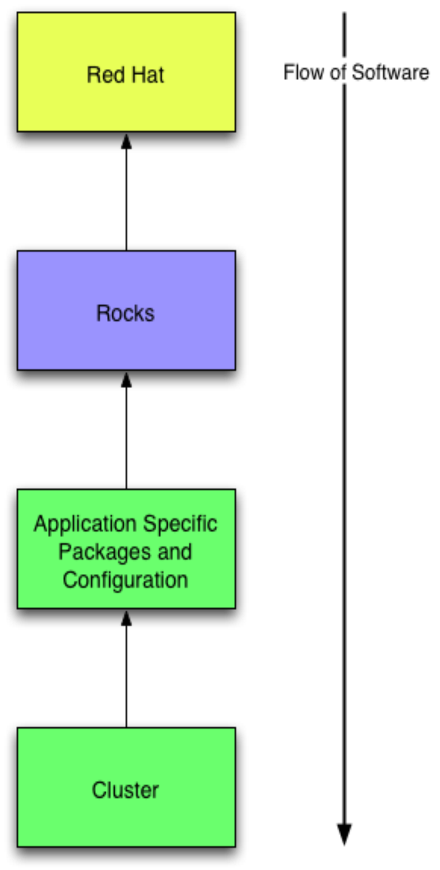
How You Create Your Own Rocks

- Rocks-dist
  - Merges all RPMs
    - Rocks
    - Yours
  - Resolves versions
  - Creates Rocks++
- Your distribution
  - Looks just like Rocks
  - Application optimized Rocks

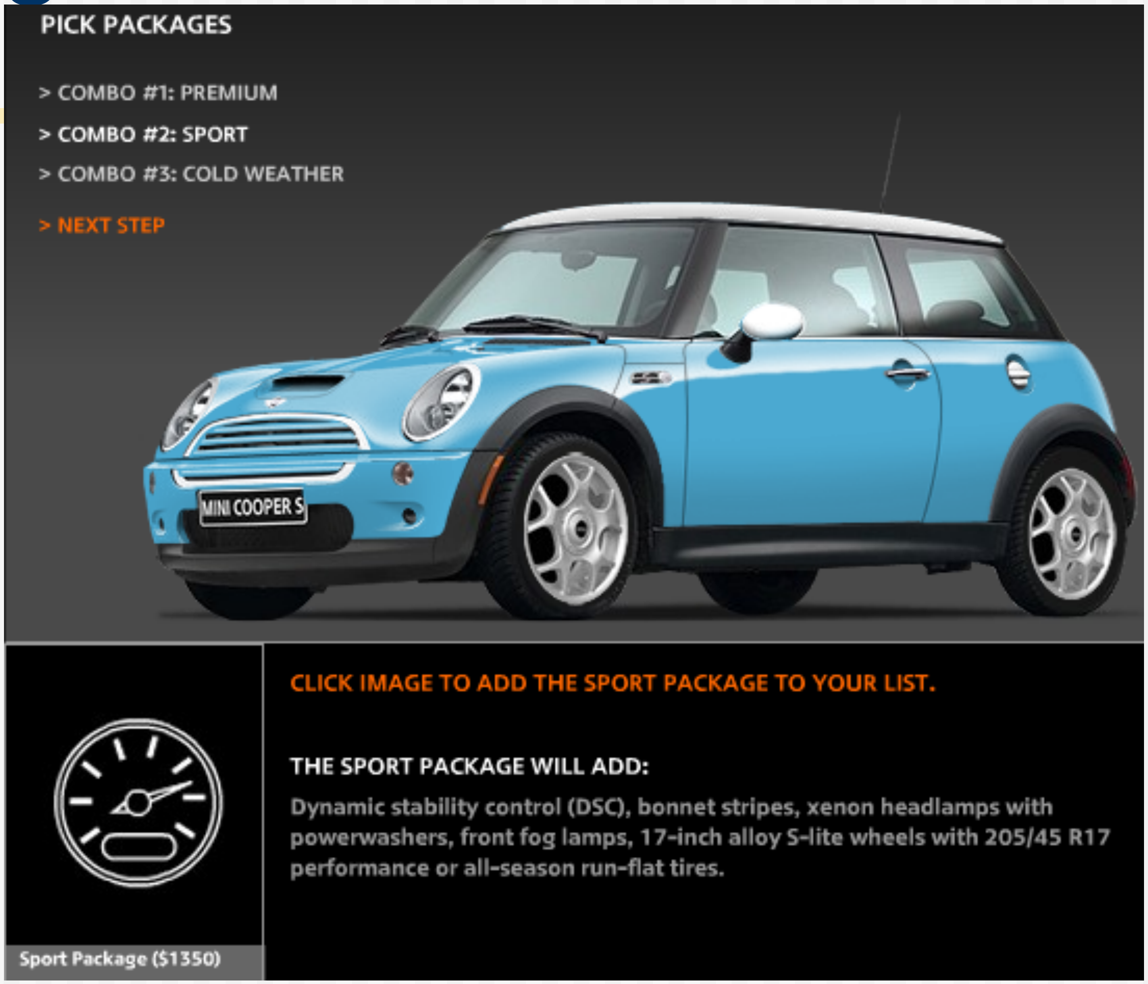


Extension Through Inheritance

- UCSD/SDSC Rocks
  - BIRN
  - GAMESS Portal
  - GEON
  - GriPhyN
  - Camera
  - Optiputer
- Commercial
  - Scalable Systems
  - Platform Computing
- Can also override existing functionality
  - Rocks without NFS?
  - Rocks for the desktop?



Rolls



CLICK IMAGE TO ADD THE SPORT PACKAGE TO YOUR LIST.

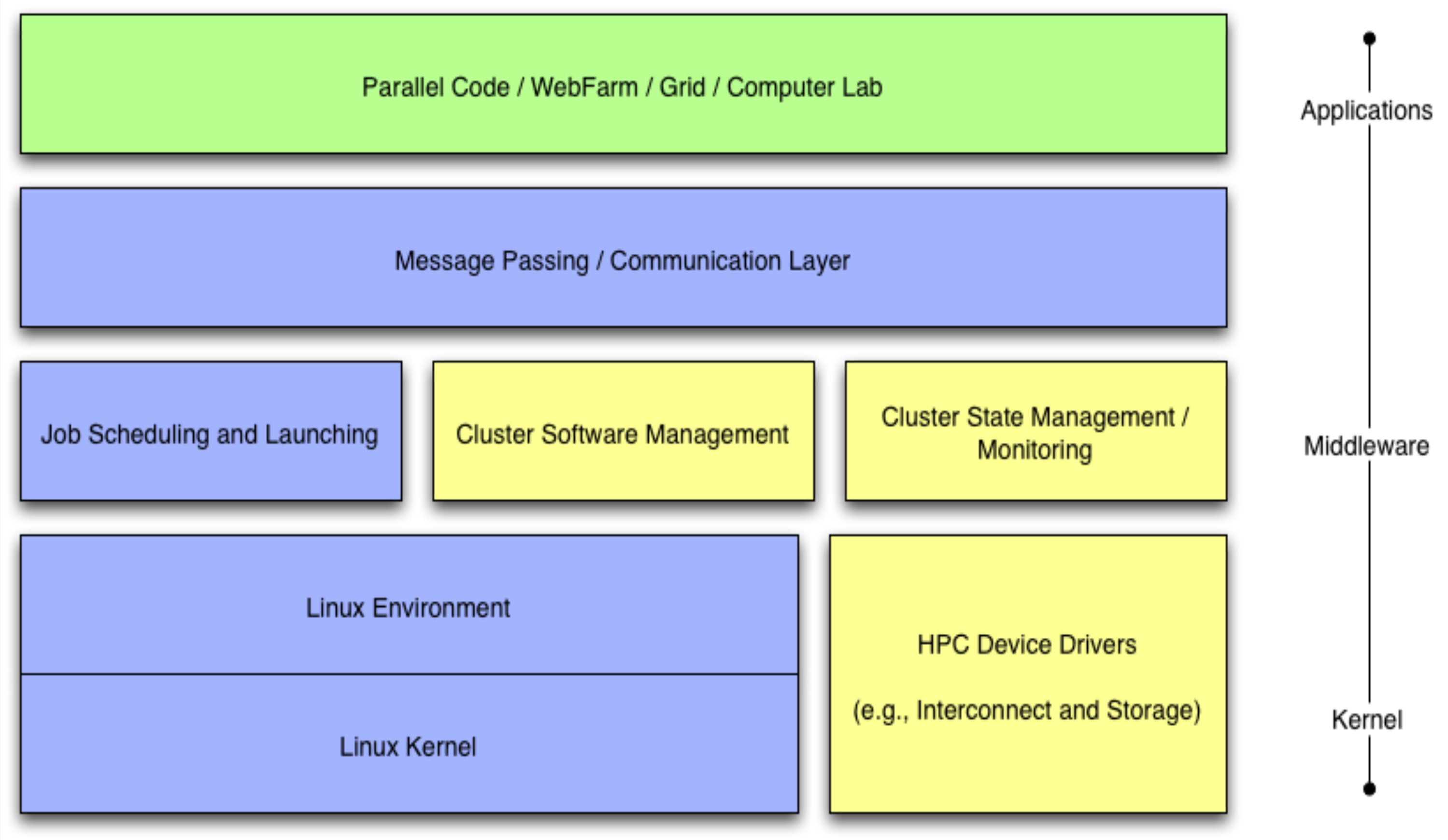
THE SPORT PACKAGE WILL ADD:

Dynamic stability control (DSC), bonnet stripes, xenon headlamps with powerwashers, front fog lamps, 17-inch alloy S-lite wheels with 205/45 R17 performance or all-season run-flat tires.

Sport Package ($1350)

- Think of a roll as a “package” for a car

Rolls Break Apart Rocks


Parallel Code / WebFarm / Grid / Computer Lab
Message Passing / Communication Layer
Job Scheduling and Launching
Cluster Software Management
Cluster State Management / Monitoring
Linux Environment
Linux Kernel
HPC Device Drivers
(e.g., Interconnect and Storage)
Applications
Middleware
Kernel

Rocks is What You Make it

- Motivation
  - “I’m concerned Rocks is becoming everything for everyone” - rocks mailing list
  - “Building a cluster should be like ordering a car. I want the sports package, but not the leather seats, …” - z4 owning rocks developer
  - We need to let go of Rocks but hold onto the core
    - Recruit more external open-source developers
    - Only trust ourselves with fundamental architecture and implementation
  - We wanted to move the SGE but need to still support PBS
- Rolls
  - Optional configuration and software
  - Just another CD for installed (think application pack)
  - SGE and PBS are different Rolls
    - User chooses scheduler
    - PBS Roll supported by Norway
    - SGE Roll supported by Singapore (and us)
  - Rolls give us more flexibility and less work
- Rocks is done
  - The core is basically stable and needs continued support
  - Rolls allow us to develop new ideas
  - Application Domain specific
- IEEE Cluster 2004 - “Rolls: Modifying a Standard System Installer to Support User-Customizable Cluster Frontend Appliances”

Extensible Rocks

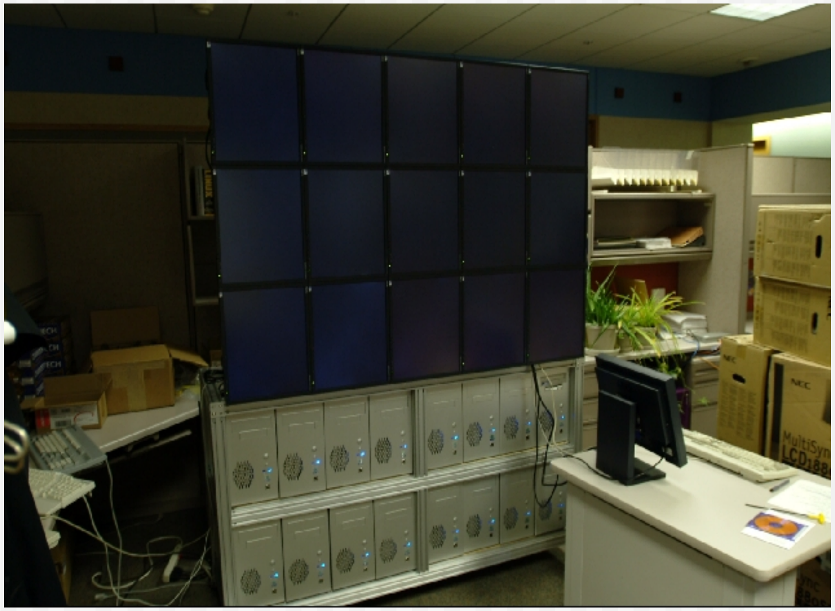
- Over a dozen Rolls already created (e. g.)
  - SGE, PBS
  - Grid (NMI stack)
  - Java
  - Condor
  - SCE
- Several third party Rolls have started
  - Quadrics (rumored)
  - PGI (just Released)
  - NIMROD
  - BIRN
  - DB2
- Rocks is done
  - The core is basically stable and needs continued support
  - Rolls allow us to develop new ideas
  - Application Domain specific
  - For example: Visualization…

Viz Roll

Rocks becomes more than just compute clusters

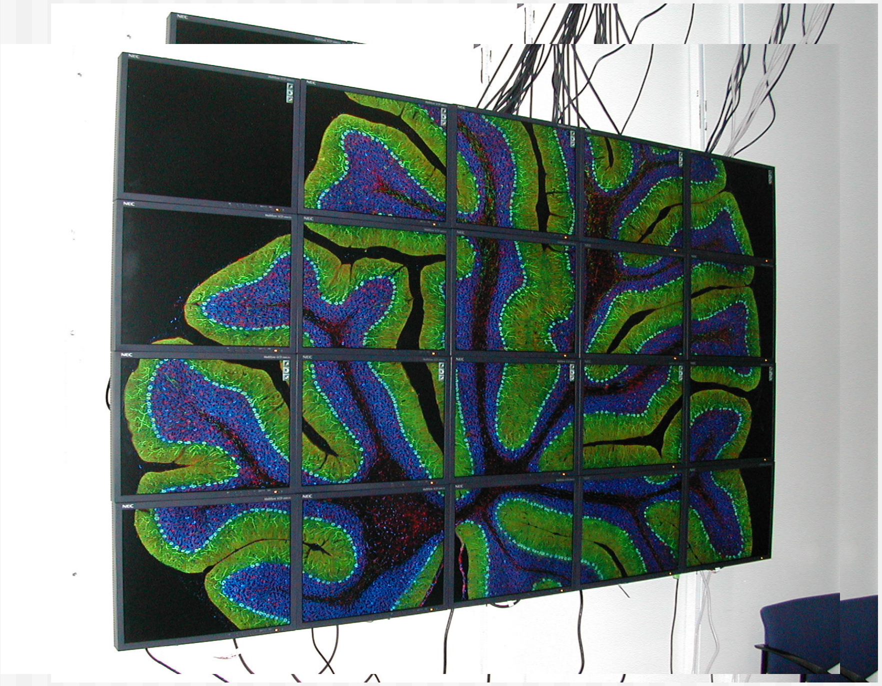
Early Work: NCSA

- LCD Cluster
  - Custom framing
  - One PC / tile
  - Portable (luggable)
  - SC 2001 Demo
- NCSA Software
  - Pixel Blaster
  - Display Wall In-A-Box
  - OSCAR based
  - Never fully released



NCMIR

- Using Rocks
- Hand configured a visualization cluster
- “Administered the machine to the point of instability” - David Lee
- Automation is needed



COTS Vis: GeoWall

- LCD Clusters
  - One PC / tile
  - Gigabit Ethernet
  - Optional Stereo Glasses
  - Portable
  - Commercial Frame (Reason)
- Applications
  - Large remote sensing
  - Volume Rendering
  - Seismic Interpretation
  - Brain mapping (NCMIR)
- Electronic Visualization Lab
  - Jason Leigh (UIC)

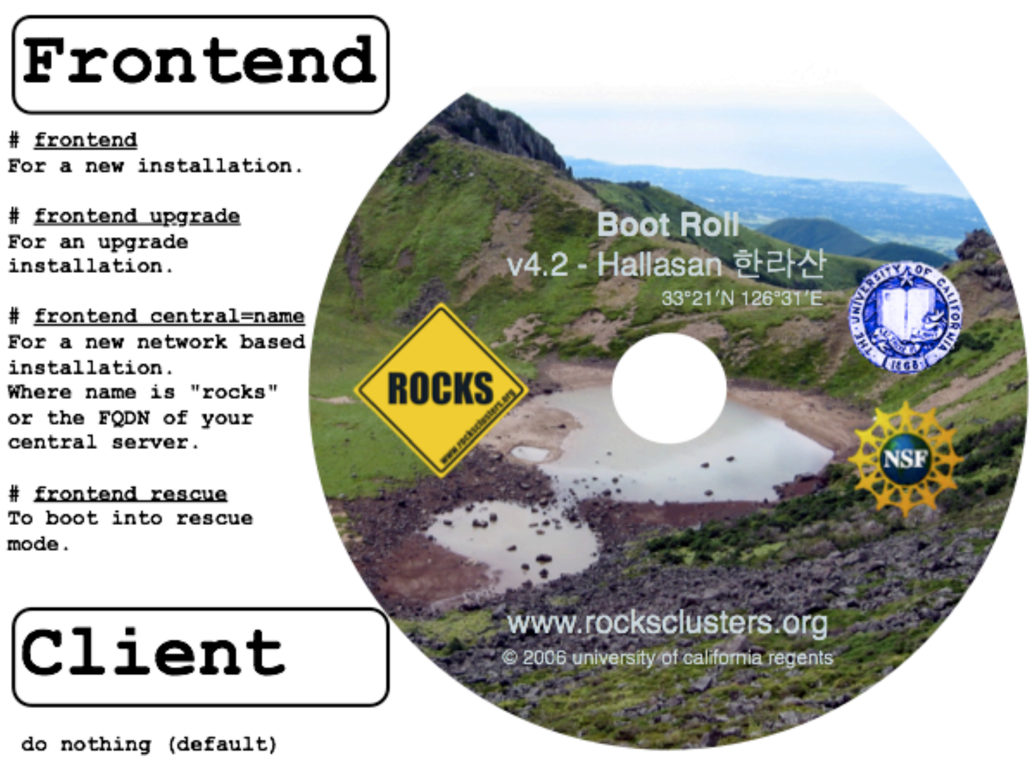
Eye Candy (NCMIR)

Rocks Installation

Step by step instruction for building your cluster

Frontend Installation

- Turn on node
- Insert CDROM
- Type
  - `frontend`



Frontend

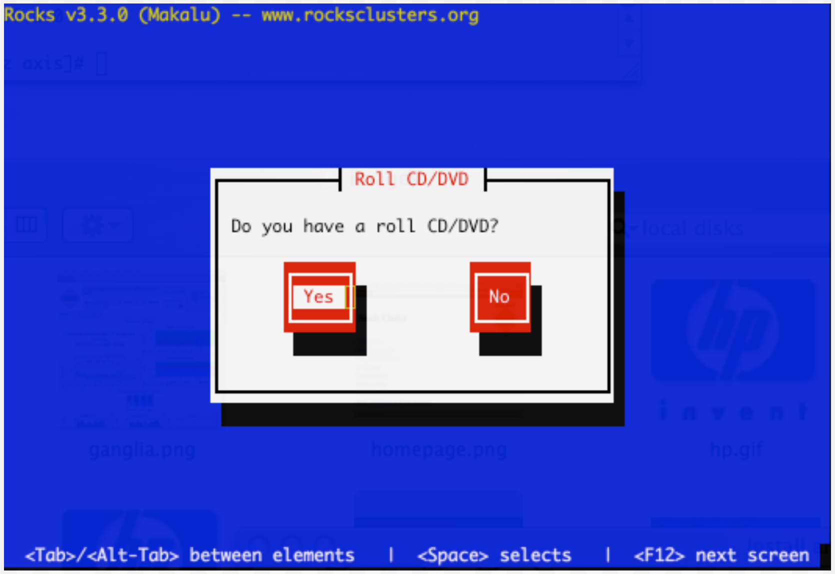
```
# frontend
For a new installation.

# frontend upgrade
For an upgrade
installation.

# frontend central=name
For a new network based
installation.
Where name is "rocks"
or the FQDN of your
central server.

# frontend rescue
To boot into rescue
mode.
```

Client

```
do nothing (default)
```

Rolls

- Anaconda Starts
- Asks for Rolls
- Select “Yes”
- Insert
  - base
  - hpc+kernel
  - area51+java+grid+sge



Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

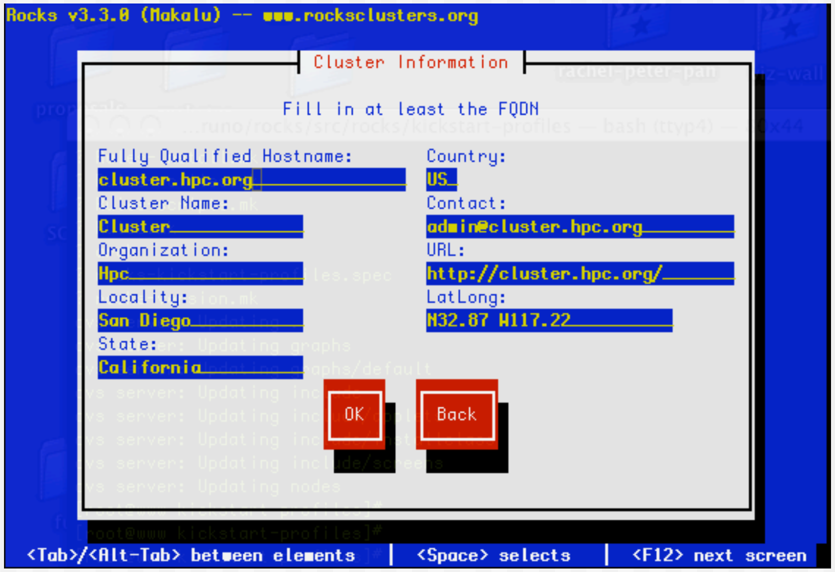
Roll CD/DVD

Do you have a roll CD/DVD?

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

Cluster Information

- Specific to Rocks
- Used for Certificates
  - SSL/HTTPS
  - Globus
- Hostname
  - Must be FQDN
  - Must be in DNS
  - Must not be an Alias



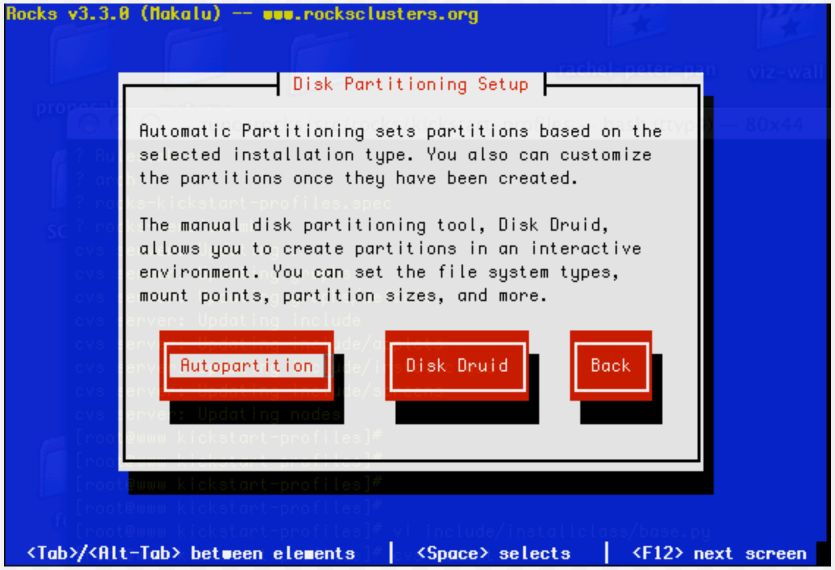
```
Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

                 Cluster Information

            Fill in at least the FQDN

Fully Qualified Hostname:        Country:
cluster.hpc.org                  US
Cluster Name:                    Contact:
Cluster                          admin@cluster.hpc.org
Organization:                    URL:
Hpc                              http://cluster.hpc.org/
Locality:                        LatLong:
San Diego                        N32.87 W117.22
State:
California
```

```
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen
```

© 2006 UC Regents

Partitioning

- Automatic
  - 6GB /
  - 1GB swap
  - Remainder for /export
- Manual
  - You choose
  - Must create a /export
- Select Wisely



Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

Disk Partitioning Setup

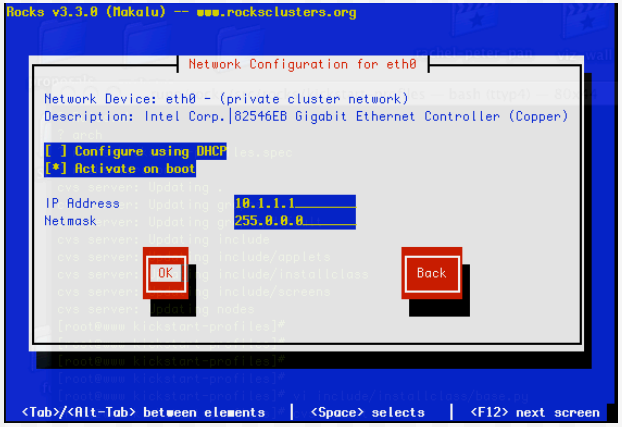
Automatic Partitioning sets partitions based on the selected installation type. You also can customize the partitions once they have been created.

The manual disk partitioning tool, Disk Druid, allows you to create partitions in an interactive environment. You can set the file system types, mount points, partition sizes, and more.

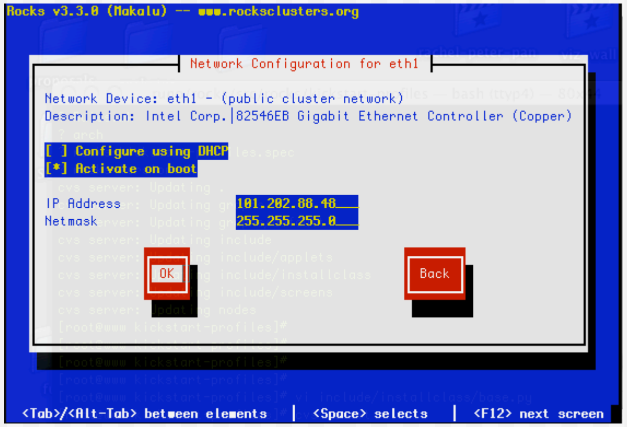
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

Networks

- Private Network
  - eth0
  - Cluster-side only
- Public Network
  - eth1
  - Internet/LAN side
- You must configure both and have 2 NICs



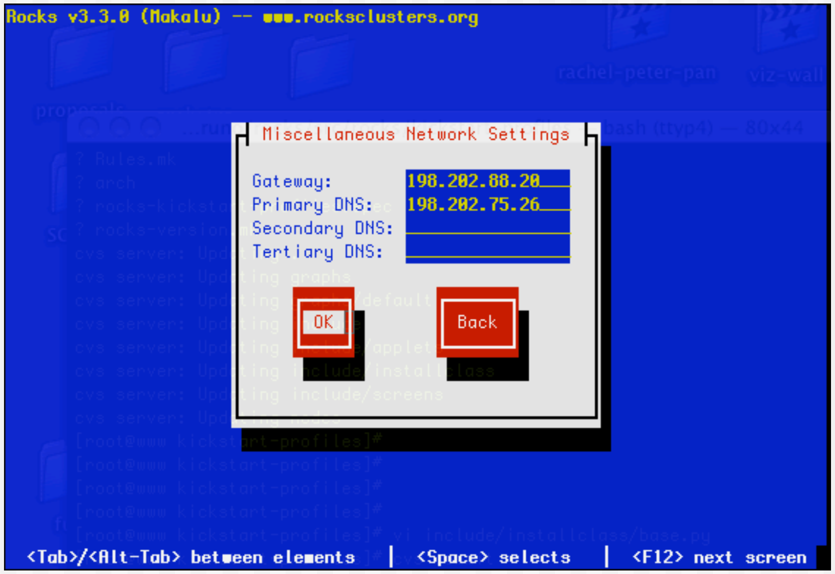
```
Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

Network Configuration for eth0

Network Device: eth0 - (private cluster network)
Description: Intel Corp.|82546EB Gigabit Ethernet Controller (Copper)

[ ] Configure using DHCP
[*] Activate on boot

IP Address   10.1.1.1________
Netmask      255.0.0.0_______

<Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```



```
Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

Network Configuration for eth1

Network Device: eth1 - (public cluster network)
Description: Intel Corp.|82546EB Gigabit Ethernet Controller (Copper)

[ ] Configure using DHCP
[*] Activate on boot

IP Address   101.202.88.48___
Netmask      255.255.255.0___

<Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

Gateway

- Gateway / DNS
  - Same as any other device on you network
- All traffic for compute nodes is NATed through the frontend.
- DNS is only for the frontend, compute nodes use the frontend as their DNS.



Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

Miscellaneous Network Settings

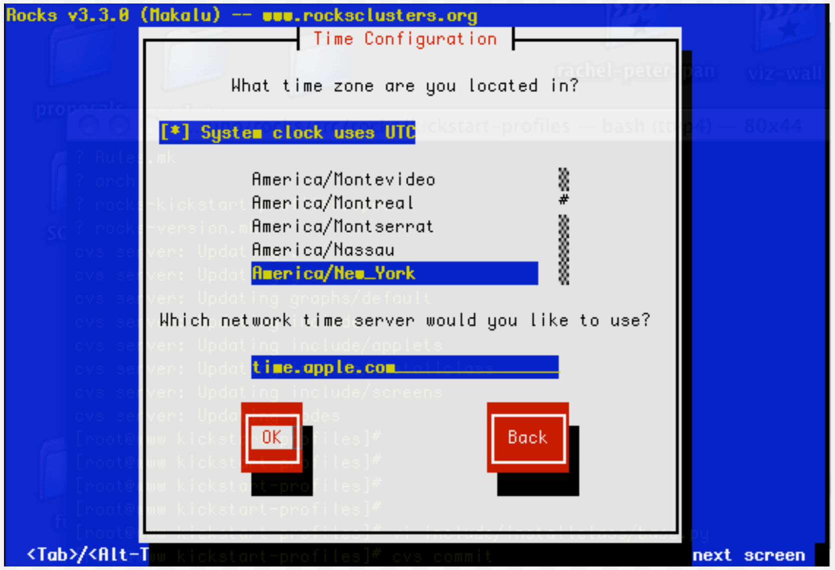
```
Gateway:        198.202.88.20
Primary DNS:    198.202.75.26
Secondary DNS:
Tertiary DNS:
```

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

© 2006 UC Regents

Network Time Protocol

- Choose timezone
  - UTC is a good choice
  - Or localize
- Default server is
  - time.apple.com
  - Change it if you wish



Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

Time Configuration

What time zone are you located in?

[*] System clock uses UTC

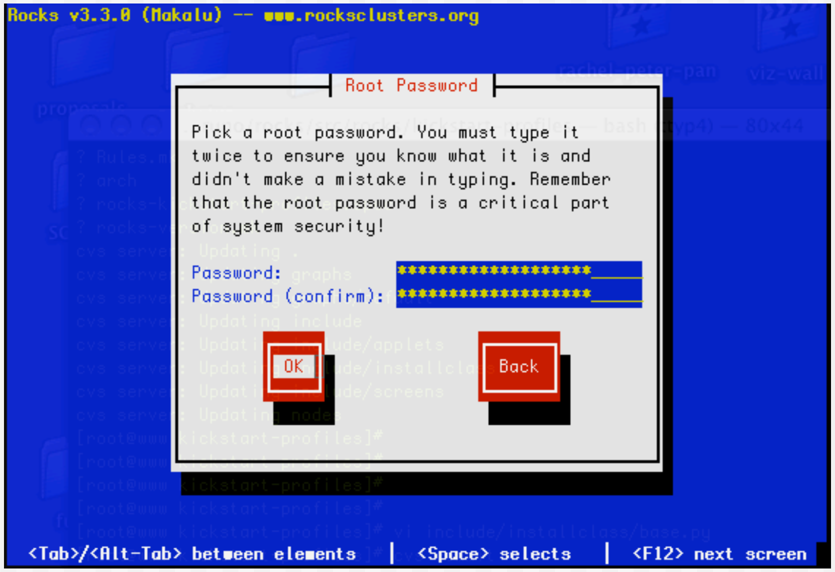
```
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
```

Which network time server would you like to use?

time.apple.com

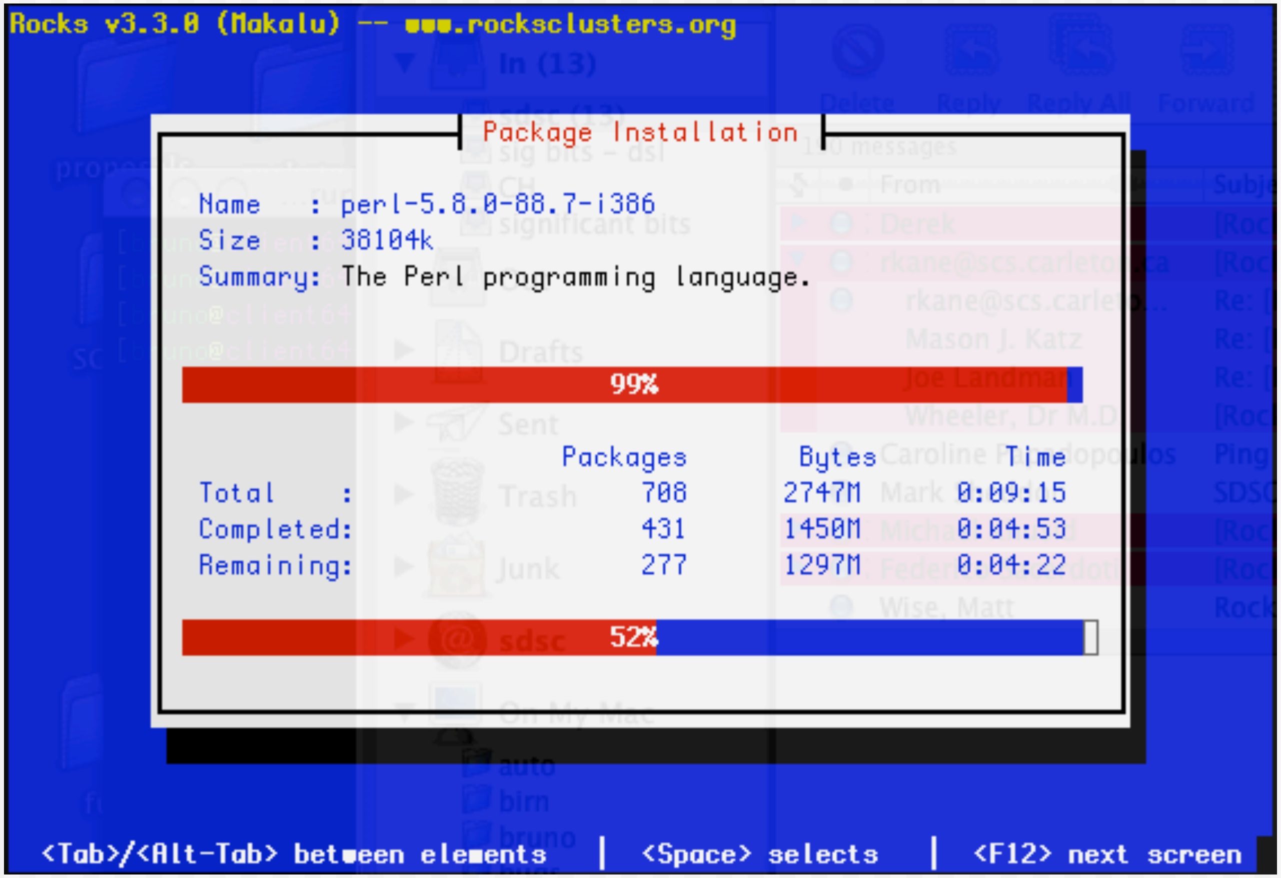
Root Password

- Password is secure
  - Not stored in clear text form anywhere (not in DB)
- Also used for mysql password



```
Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

              Root Password

   Pick a root password. You must type it
   twice to ensure you know what it is and
   didn't make a mistake in typing. Remember
   that the root password is a critical part
   of system security!

   Password:            ********************
   Password (confirm):  ********************

          OK                    Back

 <Tab>/<Alt-Tab> between elements  |  <Space> selects  |  <F12> next screen
```

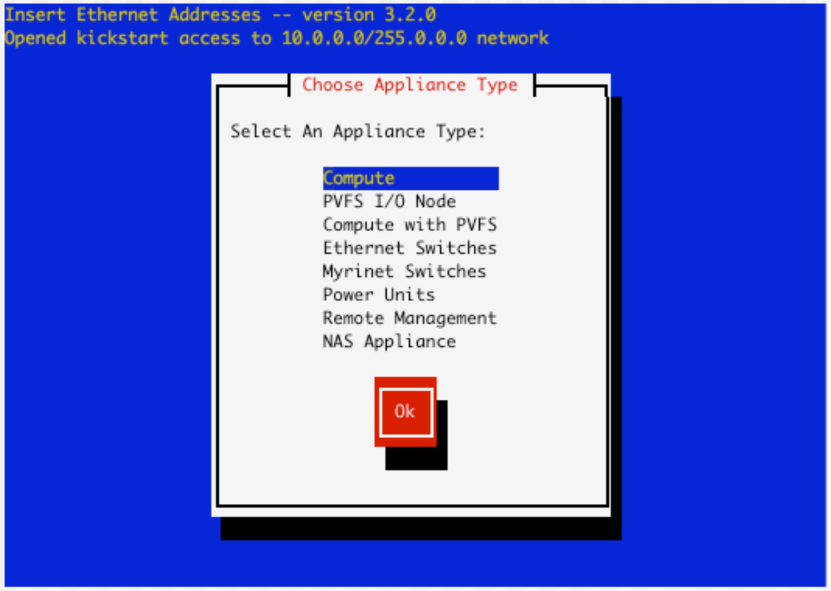
Installing Packages



```
Rocks v3.3.0 (Makalu) -- www.rocksclusters.org

                    Package Installation

  Name    : perl-5.8.0-88.7-i386
  Size    : 38104k
  Summary: The Perl programming language.

                          99%

                    Packages        Bytes       Time
  Total     :            708        2747M    0:09:15
  Completed:             431        1450M    0:04:53
  Remaining:             277        1297M    0:04:22

                          52%

 <Tab>/<Alt-Tab> between elements  |  <Space> selects  |  <F12> next screen
```

Integrate Compute Nodes

- Log into Frontend (as root)
- Run `insert-ethers`
  - Can choose appliance type
  - Rolls add new appliance types
  - For now we will use Compute
- Turn on first node
  - Nodes are integrated serially
  - Need to map machine name to machine location
  - After we integrate machines can be re-installed in parallel
- Remote Terminal (ekv)
  - `ssh compute-0-0 -p2200`

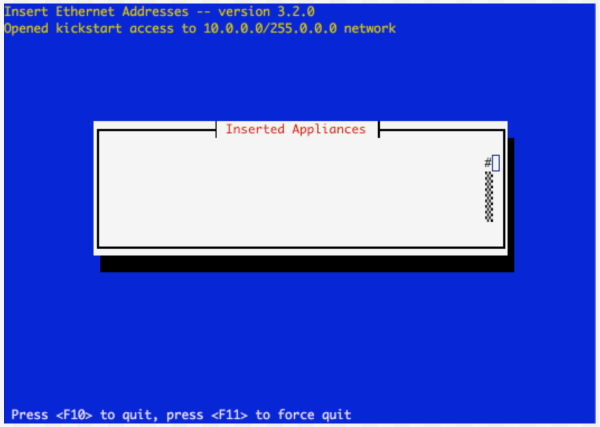


```
Insert Ethernet Addresses -- version 3.2.0
Opened kickstart access to 10.0.0.0/255.0.0.0 network
```

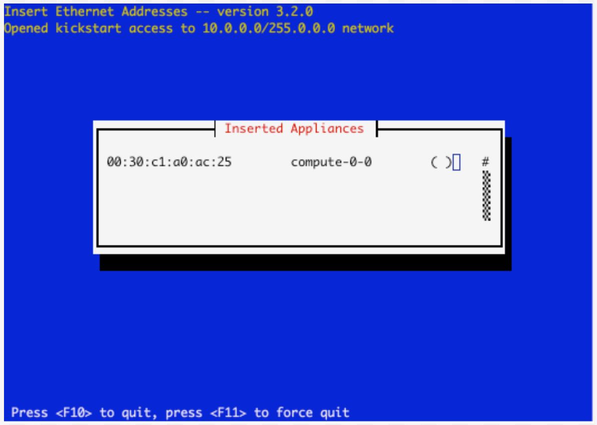
Choose Appliance Type

Select An Appliance Type:

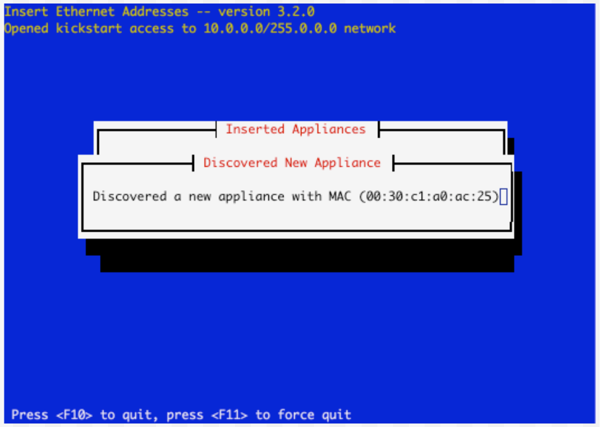
Discovering Compute-0-0



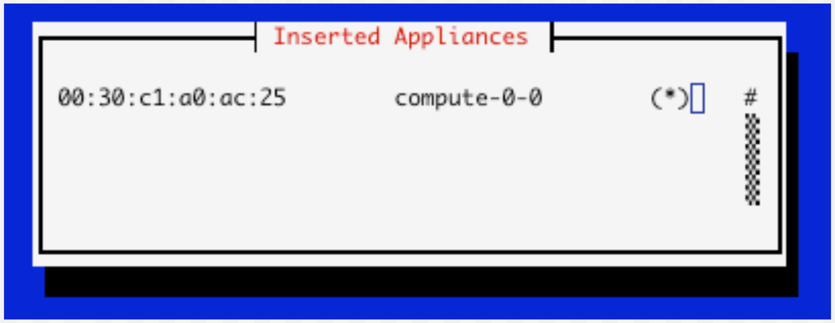
```
Insert Ethernet Addresses -- version 3.2.0
Opened kickstart access to 10.0.0.0/255.0.0.0 network

                    ┤ Inserted Appliances ├
                                                       #

Press <F10> to quit, press <F11> to force quit
```



```
Insert Ethernet Addresses -- version 3.2.0
Opened kickstart access to 10.0.0.0/255.0.0.0 network

                    ┤ Inserted Appliances ├
   00:30:c1:a0:ac:25         compute-0-0          ( )    #

Press <F10> to quit, press <F11> to force quit
```



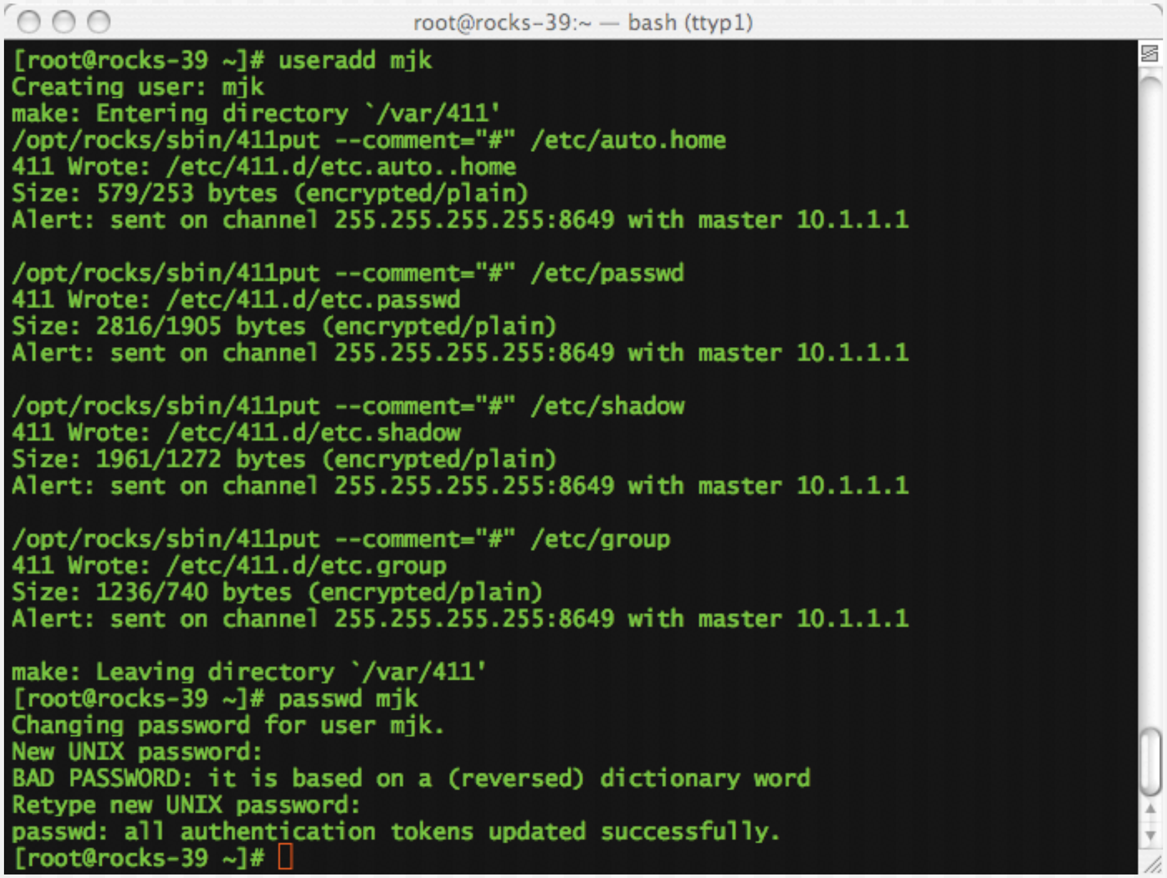
```
Insert Ethernet Addresses -- version 3.2.0
Opened kickstart access to 10.0.0.0/255.0.0.0 network

                    ┤ Inserted Appliances ├
                 ┤ Discovered New Appliance ├
   Discovered a new appliance with MAC (00:30:c1:a0:ac:25)

Press <F10> to quit, press <F11> to force quit
```



```
                    ┤ Inserted Appliances ├
   00:30:c1:a0:ac:25         compute-0-0          (*)    #
```

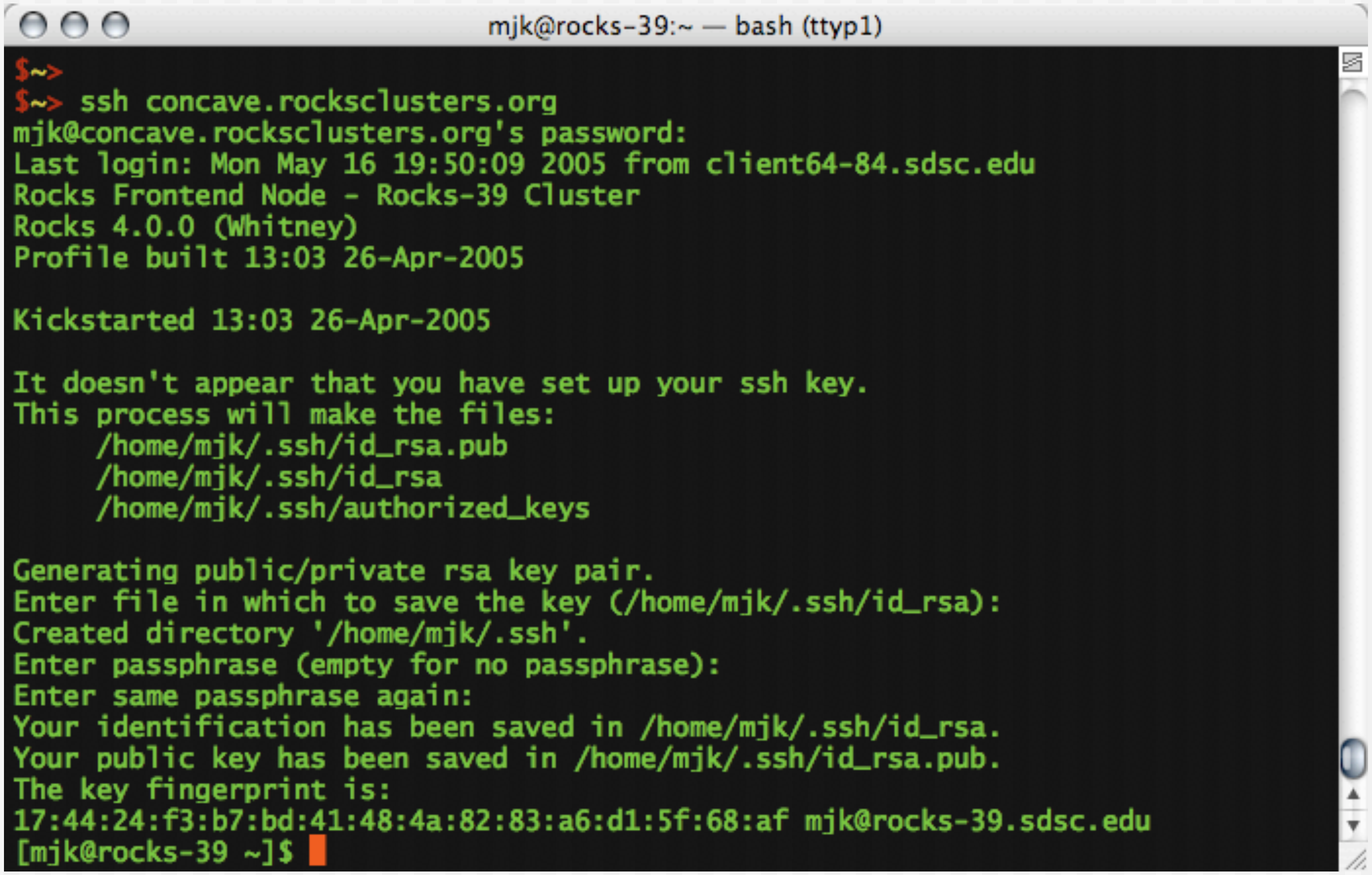
Retrieved kickstart file

useradd



```
[root@rocks-39 ~]# useradd mjk
Creating user: mjk
make: Entering directory `/var/411'
/opt/rocks/sbin/411put --comment="#" /etc/auto.home
411 Wrote: /etc/411.d/etc.auto..home
Size: 579/253 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

/opt/rocks/sbin/411put --comment="#" /etc/passwd
411 Wrote: /etc/411.d/etc.passwd
Size: 2816/1905 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

/opt/rocks/sbin/411put --comment="#" /etc/shadow
411 Wrote: /etc/411.d/etc.shadow
Size: 1961/1272 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

/opt/rocks/sbin/411put --comment="#" /etc/group
411 Wrote: /etc/411.d/etc.group
Size: 1236/740 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1

make: Leaving directory `/var/411'
[root@rocks-39 ~]# passwd mjk
Changing password for user mjk.
New UNIX password:
BAD PASSWORD: it is based on a (reversed) dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@rocks-39 ~]#
```

user login



```
$~>
$~> ssh concave.rocksclusters.org
mjk@concave.rocksclusters.org's password:
Last login: Mon May 16 19:50:09 2005 from client64-84.sdsc.edu
Rocks Frontend Node - Rocks-39 Cluster
Rocks 4.0.0 (Whitney)
Profile built 13:03 26-Apr-2005

Kickstarted 13:03 26-Apr-2005

It doesn't appear that you have set up your ssh key.
This process will make the files:
     /home/mjk/.ssh/id_rsa.pub
     /home/mjk/.ssh/id_rsa
     /home/mjk/.ssh/authorized_keys

Generating public/private rsa key pair.
Enter file in which to save the key (/home/mjk/.ssh/id_rsa):
Created directory '/home/mjk/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mjk/.ssh/id_rsa.
Your public key has been saved in /home/mjk/.ssh/id_rsa.pub.
The key fingerprint is:
17:44:24:f3:b7:bd:41:48:4a:82:83:a6:d1:5f:68:af mjk@rocks-39.sdsc.edu
[mjk@rocks-39 ~]$
```

End