



# Introduction to Clusters and Rocks Overview

---

## Rocks for Noobs



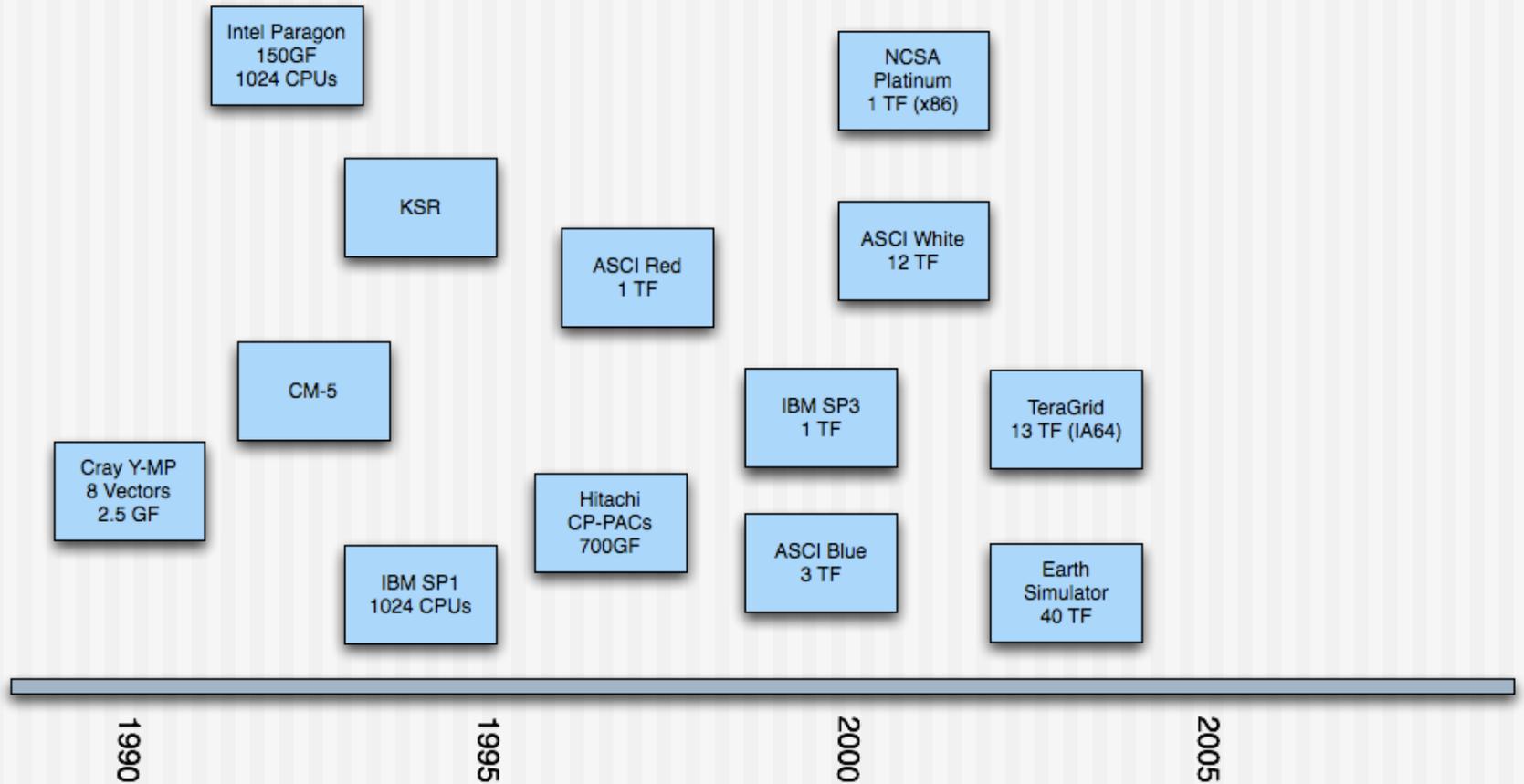
# Outline

---

- ◆ History of Clusters
- ◆ Lesson on Optimization
- ◆ Rocks
  - ⇒ Philosophies
  - ⇒ Components
  - ⇒ Architecture
- ◆ Complex Computing Infrastructures
  - ⇒ Visualization Walls
  - ⇒ Bio-Informatic Systems

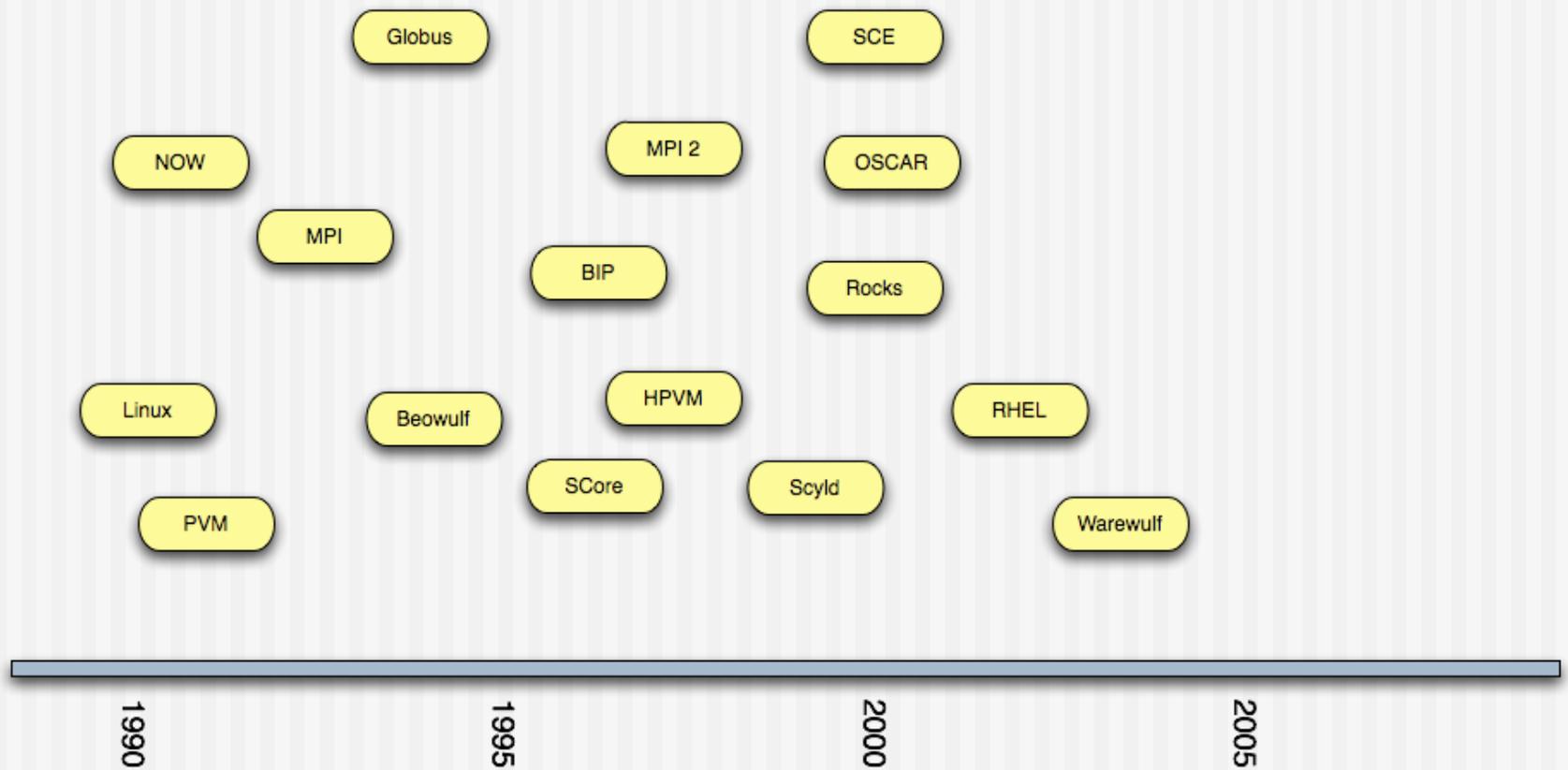


# Sampling of High Performance Computing (HPC) Hardware



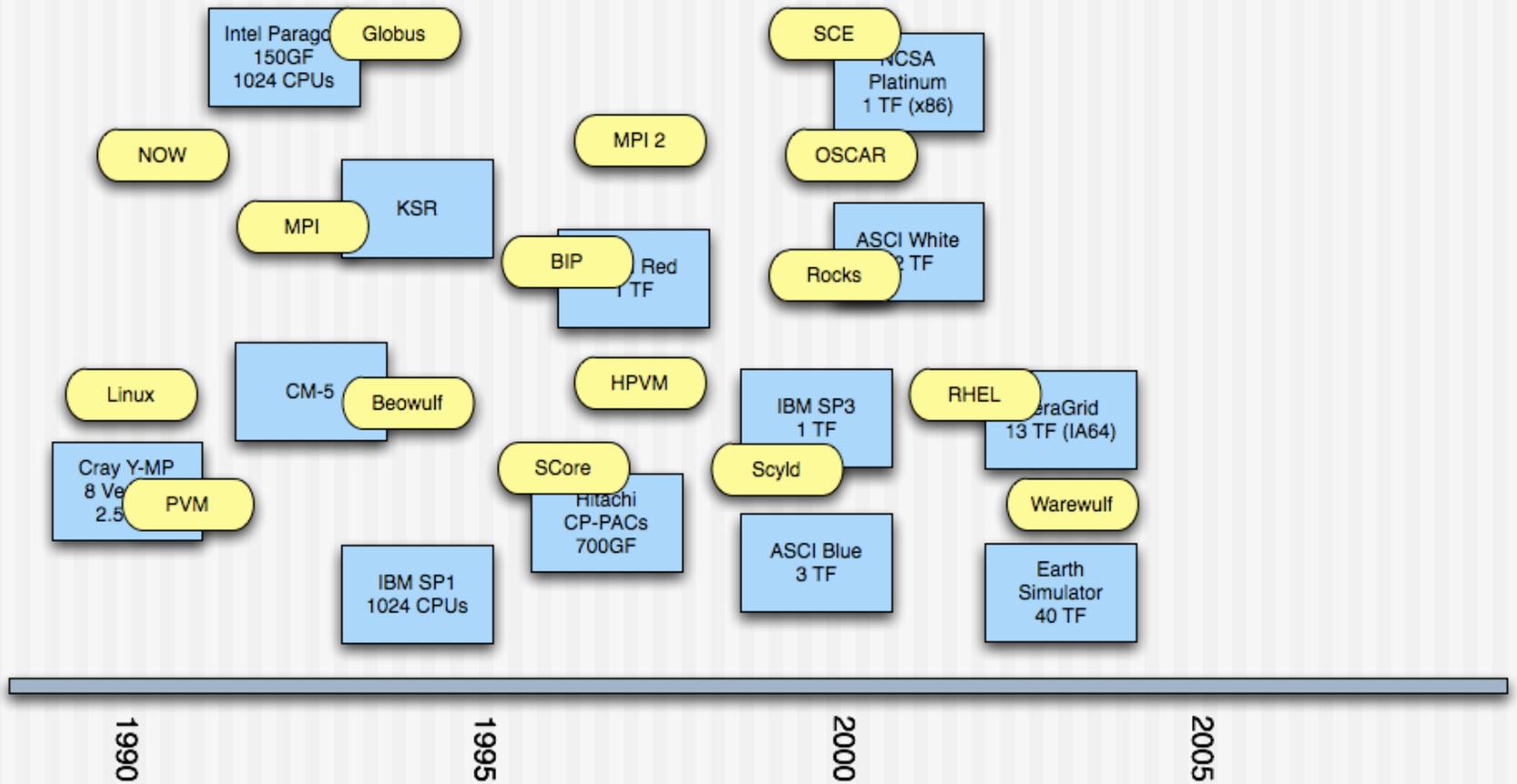


# Some Significant Software





# Relationships





# NOW

## Network of Workstations

- ◆ Pioneered the vision for clusters of commodity processors.
  - David Culler (UC Berkeley) started early 90's
  - SunOS on SPARC Microprocessor
  - High Performance, Low Latency Interconnect
    - First generation of Myrinet
    - Active Messages
  - Glunix (Global Unix) execution environment
- ◆ Brought key issues to the forefront of commodity-based computing
  - Global OS
  - Parallel file systems
  - Fault tolerance
  - High-performance messaging
  - System Management



# Beowulf

[www.beowulf.org](http://www.beowulf.org)

---

- ◆ Definition
  - ⇒ Collection of commodity computers (PCs)
  - ⇒ Using a commodity network (Ethernet)
  - ⇒ Running open-source operating system (Linux)
- ◆ Interconnect
  - ⇒ Gigabit Ethernet (commodity)
    - High Latency
    - Cheap
  - ⇒ Myrinet, Infiniband, ... (non-commodity)
    - Low Latency
    - OS-bypass
    - Expensive
  - ⇒ Programming model is Message Passing
- ◆ NOW pioneered the vision for clusters of commodity processors.
- ◆ Beowulf popularized the notion and made it very affordable
- ◆ Come to mean any Linux cluster



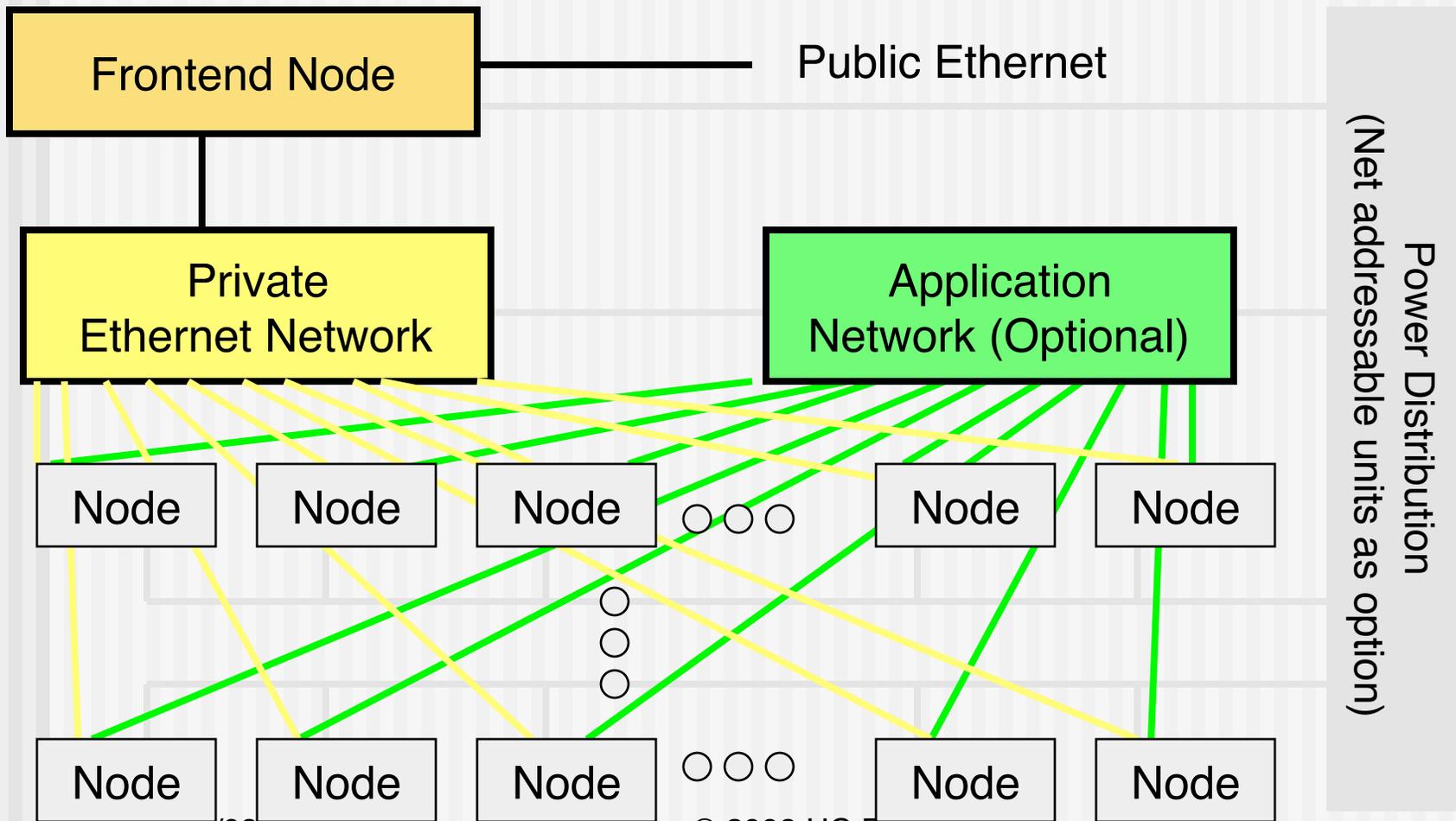
# Outcomes of NOW / Beowulf

---

- ◆ Clusters of PCs Popularized
- ◆ Allowed more people to work on parallel computing
- ◆ Almost all software components published as open-source
- ◆ Brought key ingredients of MPPs into the commodity space
  - ⇒ Message passing environments
  - ⇒ Batch processing systems
- ◆ Extremely hard to build and run



# High Performance Computing Cluster



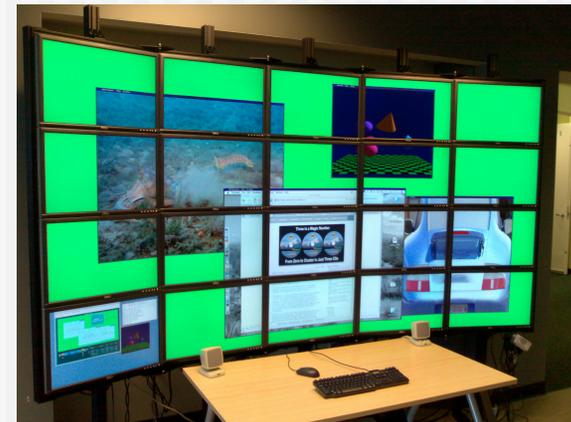
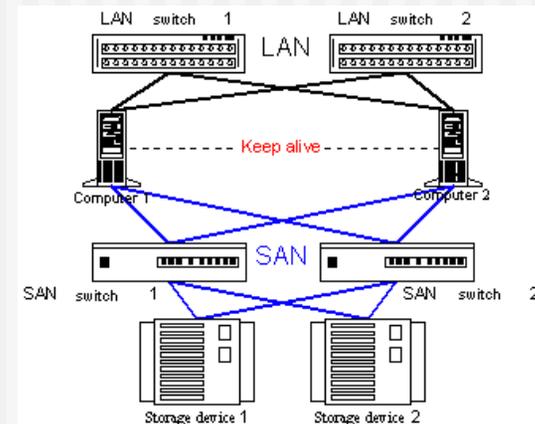
5/15/08

© 2008 UC Regents



# Other Clusters

- ◆ Highly Available (HA)
  - Generally small, less than 8 nodes
  - Redundant components
  - Multiple communication paths
  - This is not Rocks
- ◆ Visualization Clusters
  - Each node drives a display
  - OpenGL machines
  - This is not core Rocks
  - But, there is a Viz Roll





# The Dark Side of Clusters

---

- ◆ Clusters are phenomenal price/performance computational engines
  - ...
  - Can be hard to manage without experience
  - High-performance I/O is still unsolved
  - Finding out where something has failed increases at least linearly as cluster size increases
- ◆ Not cost-effective if every cluster “burns” a person just for care and feeding
- ◆ Programming environment could be vastly improved
- ◆ Technology is changing very rapidly. Scaling up is becoming commonplace (128-256 nodes)



# The Top 2 Most Critical Problems

---

- ◆ The largest problem in clusters is *software skew*
  - When software configuration on some nodes is different than on others
  - Small differences (minor version numbers on libraries) can cripple a parallel program
- ◆ The second most important problem is adequate job control of the parallel process
  - Signal propagation
  - Cleanup



# Rocks

(open source clustering distribution)

[www.rocksclusters.org](http://www.rocksclusters.org)

- ◆ Technology transfer of commodity clustering to application scientists (non-technical people)
  - ⊕ “make clusters easy”
  - ⊕ Scientists can build their own supercomputers and migrate up to national centers, or international grids, as needed
  - ⊕ Supports more than just MPI machines
- ◆ Rocks is a cluster on set of CDs (or a DVD)
  - ⊕ Red Enterprise Hat Linux (open source, *de facto* standard, and **free**)
  - ⊕ Clustering software (PBS, SGE, Ganglia, GT4, ...)
  - ⊕ Highly programmatic software configuration management
- ◆ Core software technology for many UCSD projects
  - ⊕ BIRN, CTBP, EOL, GEON, NBCR, OptIPuter, CAMERA, ...
- ◆ First Software release Nov, 2000
  - ⊕ Began as an MPI cluster solution
  - ⊕ Now builds grid resources
  - ⊕ Moving towards virtualization (XEN) and other Oses (Solaris)
- ◆ Supports x86, Opteron/EM64T, and Itanium





# Simple Deployment

- ◆ Install a frontend
  1. Insert Rocks Base CD
  2. Insert Roll CDs (optional components)
  3. Answer 7 screens of configuration data
  4. Drink coffee/tea/beer (takes about 30 minutes to install)
- ◆ Install compute nodes:
  1. Login to frontend
  2. Execute insert-ethers
  3. Boot compute node with Rocks Base CD (or PXE)
  4. Insert-ethers discovers nodes
  5. Goto step 3
- ◆ Add user accounts
- ◆ Start computing

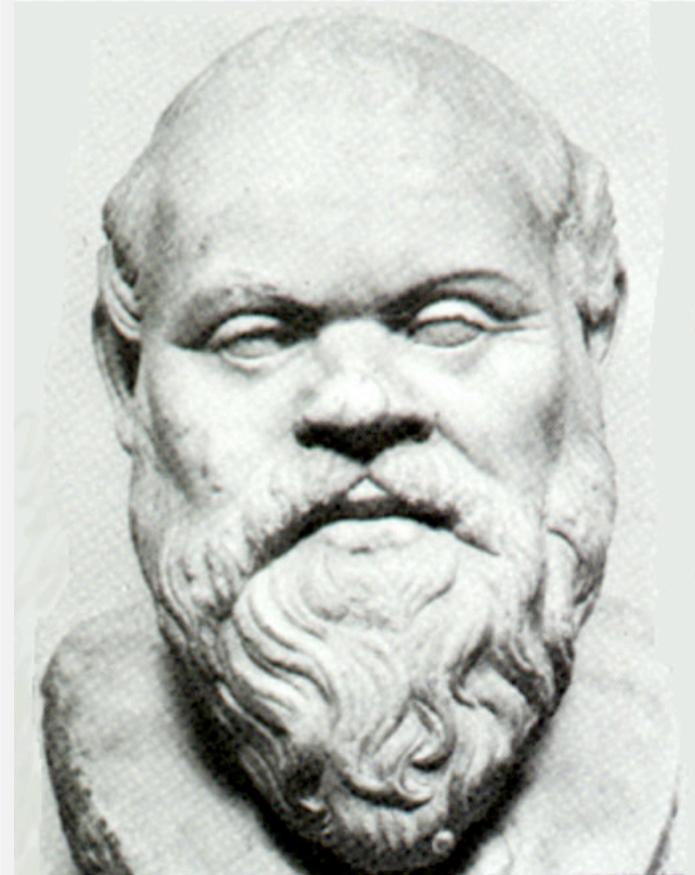


## Optional Rolls

- Condor
- Grid (based on NMI R4)
- Intel (compilers)
- Java
- SCE (developed in Thailand)
- Sun Grid Engine
- PBS (developed in Norway)
- Area51 (security monitoring tools)
- Many Others ...

# Philosophy

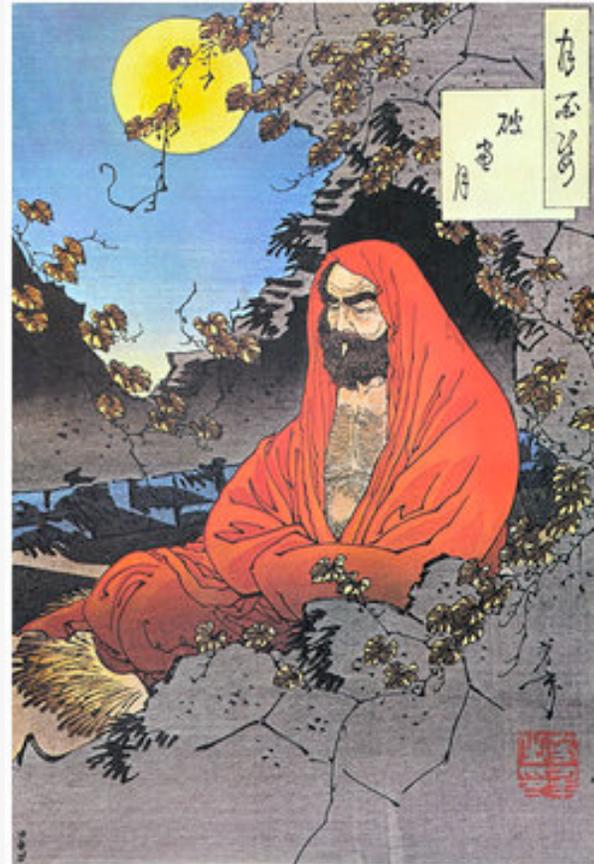
- ◆ Caring and feeding for a system is not fun
- ◆ System Administrators cost more than clusters
  - 1 TFLOP cluster is less than \$100,000 (US)
  - Close to actual cost of a fulltime administrator
- ◆ The system administrator is the weakest link in the cluster
  - Bad ones like to tinker (make small changes)
  - Good ones still make mistakes



# Philosophy

## continued

- ◆ All nodes are 100% automatically configured
  - Zero “hand” configuration
  - This includes site-specific configuration
- ◆ Run on heterogeneous standard high volume components (PCs)
  - Use components that offer the best price/performance
  - Software installation and configuration must support different hardware
  - **Homogeneous clusters do not exist**
  - Disk imaging requires homogeneous cluster



# Philosophy

## continued

- ◆ Optimize for installation
  - ⦿ Get the system up quickly
  - ⦿ In a consistent state
  - ⦿ Build supercomputers in hours not months
- ◆ Manage through re-installation
  - ⦿ Can re-install 128 nodes in under 20 minutes
  - ⦿ No support for on-the-fly system patching
- ◆ Do not spend time trying to issue system consistency
  - ⦿ Just re-install
  - ⦿ Can be batch driven
- ◆ Uptime in HPC is a myth
  - ⦿ Supercomputing sites have monthly downtime
  - ⦿ HPC is not HA (High Availability)





# Rocks Basic Approach

- ◆ Install a frontend
  1. Insert Rocks Base CD
  2. Insert Roll CDs (optional components)
  3. Answer 7 screens of configuration data
  4. Drink coffee (takes about 30 minutes to install)
- ◆ Install compute nodes:
  1. Login to frontend
  2. Execute insert-ethers
  3. Boot compute node with Rocks Base CD (or PXE)
  4. Insert-ethers discovers nodes
  5. Goto step 3
- ◆ Add user accounts
- ◆ Start computing



## Optional Rolls

- Condor
- Grid (based on NMI R4)
- Intel (compilers)
- Java
- SCE (developed in Thailand)
- Sun Grid Engine
- PBS (developed in Norway)
- Area51 (security monitoring tools)
- Many Others ...



# Minimum Requirements

---

## ◆ Frontend

- ⇒ 2 Ethernet Ports
- ⇒ CDROM
- ⇒ 18 GB Disk Drive
- ⇒ 512 MB RAM

## ◆ Compute Nodes

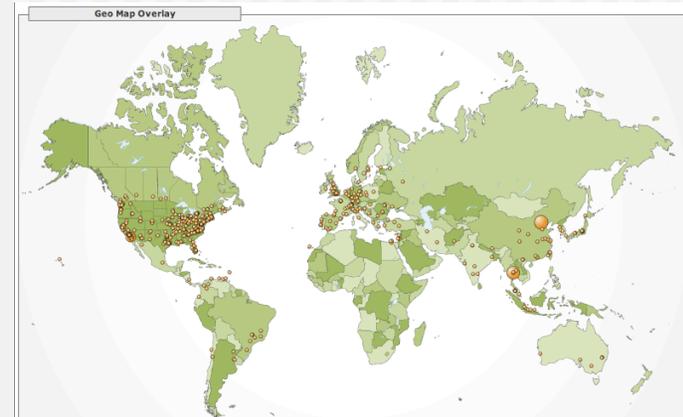
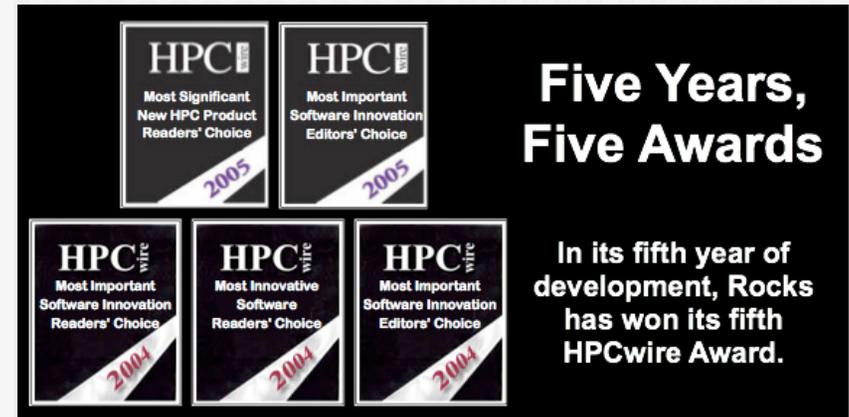
- ⇒ 1 Ethernet Port
- ⇒ 18 GB Disk Drive
- ⇒ 512 MB RAM

- Complete OS Installation on all Nodes
- No support for Diskless (yet)
- Not a Single System Image
- All Hardware must be supported by RHEL



# Rocks Users

- ◆ HPC wire awards
  - ➔ 2004, and 2005
  - ➔ Competition was commercial
- ◆ User base is international
  - ➔ Still U.S. heavy
  - ➔ Europe and Asia well represented
- ◆ High Performance Computing community is eager to adopt open-source clustering solutions





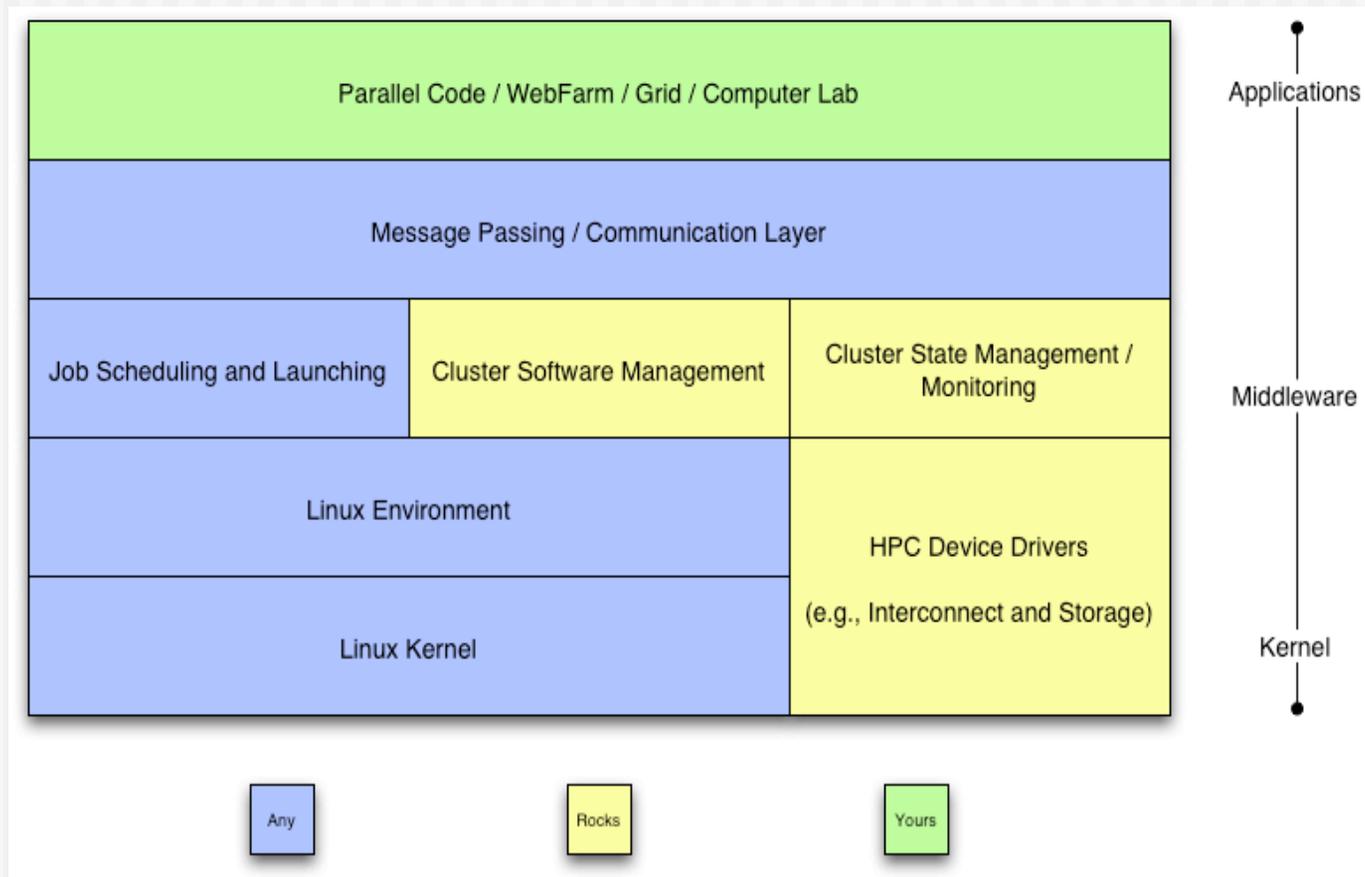
---

Making Clusters Easy

# ROCKS

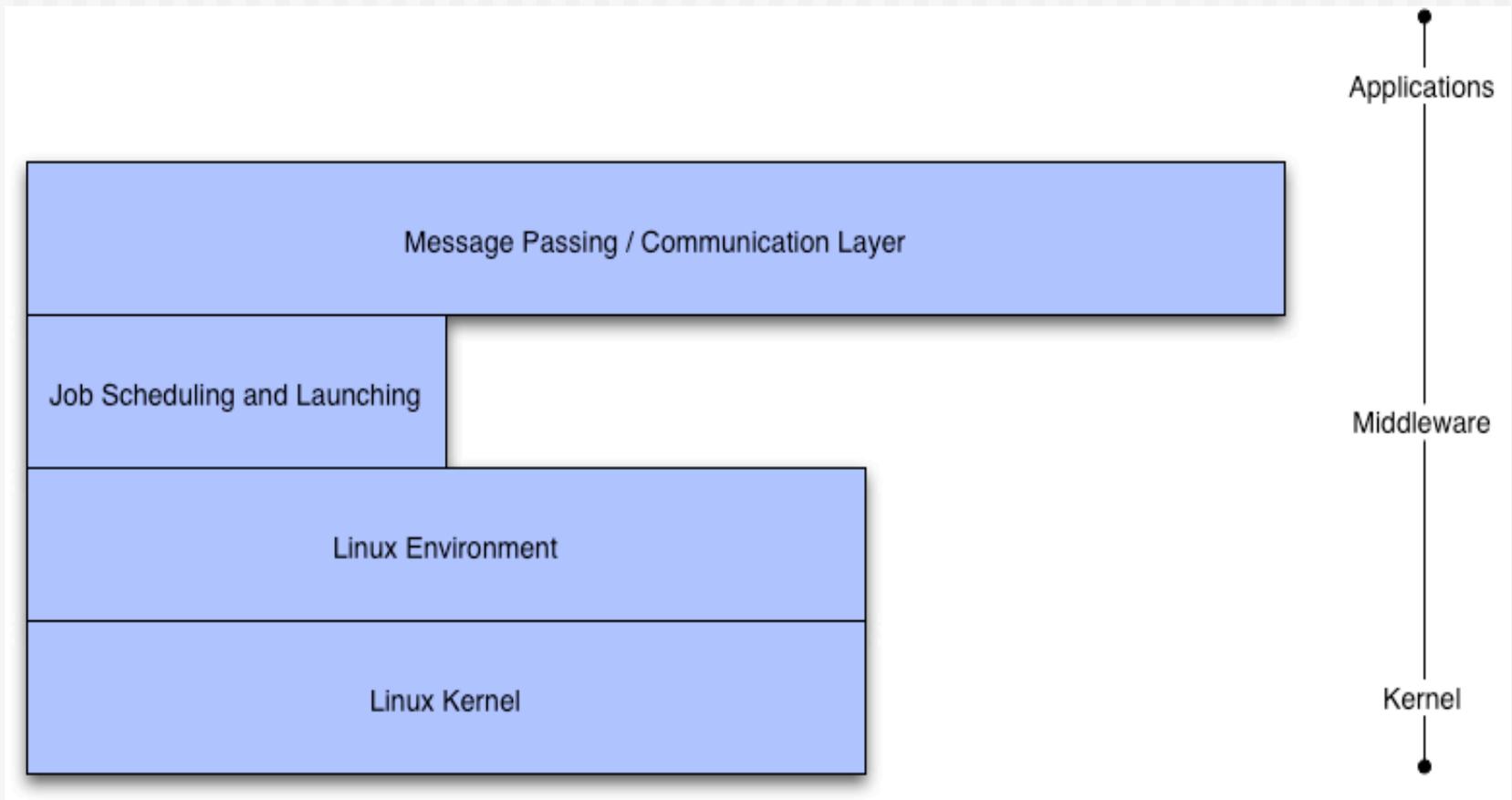


# Cluster Software Stack





# Common to Any Cluster





# 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.

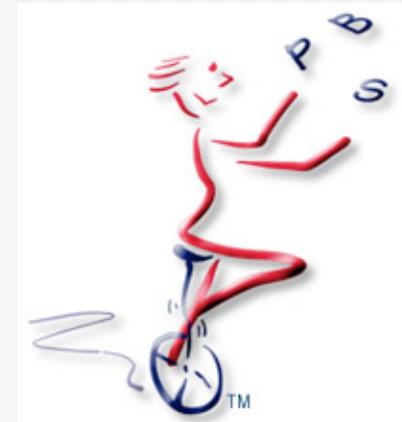


advertiser links [what's this?](#)

- [Accounting Research Manager](#)  
Find insightful interpretations on GAAP and Securities and Exchange Commission...  
[www.accountingresearchmanager.com](http://www.accountingresearchmanager.com)
- [Securities Exchange Commission](#)

# 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
  - Now the default scheduler for Rocks
  - Robust and dynamic





# Communication Layer

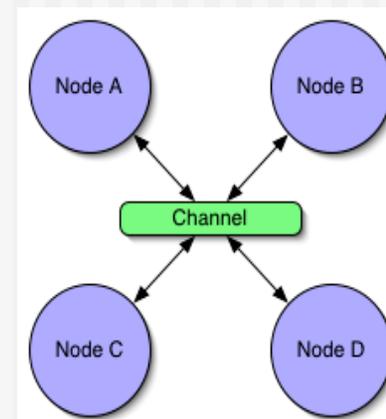
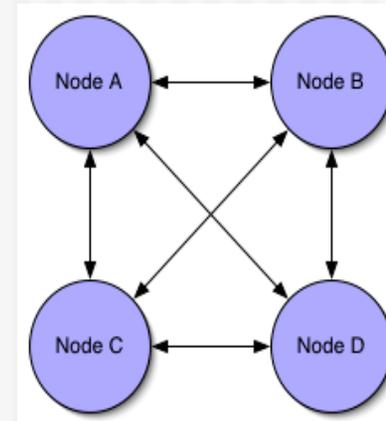
---

- ◆ 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

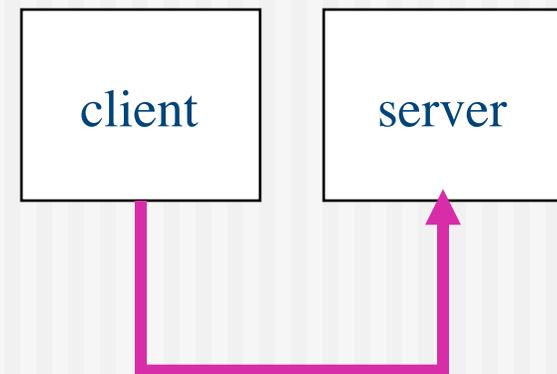
- ◆ Sockets
  - ⦿ Point-to-Point
  - ⦿  $N$  machines =  $(n^2 - n)/2$  connections
  - ⦿ 1, 3, 6, 10, 15, ...
- ◆ MPI/PVM
  - ⦿ Shared virtual channel
  - ⦿ Implementation could be sockets
  - ⦿ Easier to program





# 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

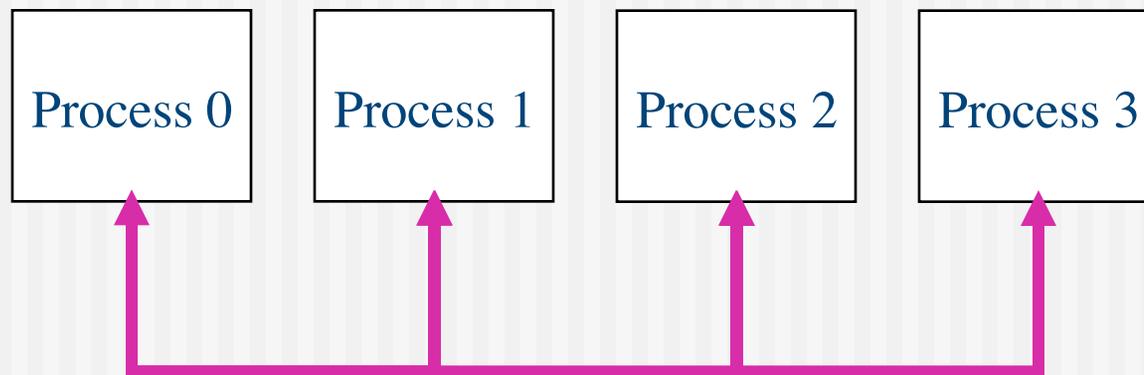
---

- ◆ 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





# 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
- ◆ Hard to manage deployment
  - ⇒ network X compiler = mpi binaries
  - ⇒ Result is several versions of MPI / cluster



# Compile

---

## ◆ MPICH with GNU Compilers and Ethernet

<b>Compiler</b>	<b>Path</b>
C:	<code>/opt/mpich/ethernet/gcc/bin/mpicc</code>
C++:	<code>/opt/mpich/ethernet/gcc/bin/mpiCC</code>
F77:	<code>/opt/mpich/ethernet/gcc/bin/mpif77</code>

## ◆ MPICH with GNU Compilers and Myrinet

<b>Compiler</b>	<b>Path</b>
C:	<code>/opt/mpich/myrinet/gcc/bin/mpicc</code>
C++:	<code>/opt/mpich/myrinet/gcc/bin/mpiCC</code>
F77:	<code>/opt/mpich/myrinet/g77/bin/mpif77</code>



# 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



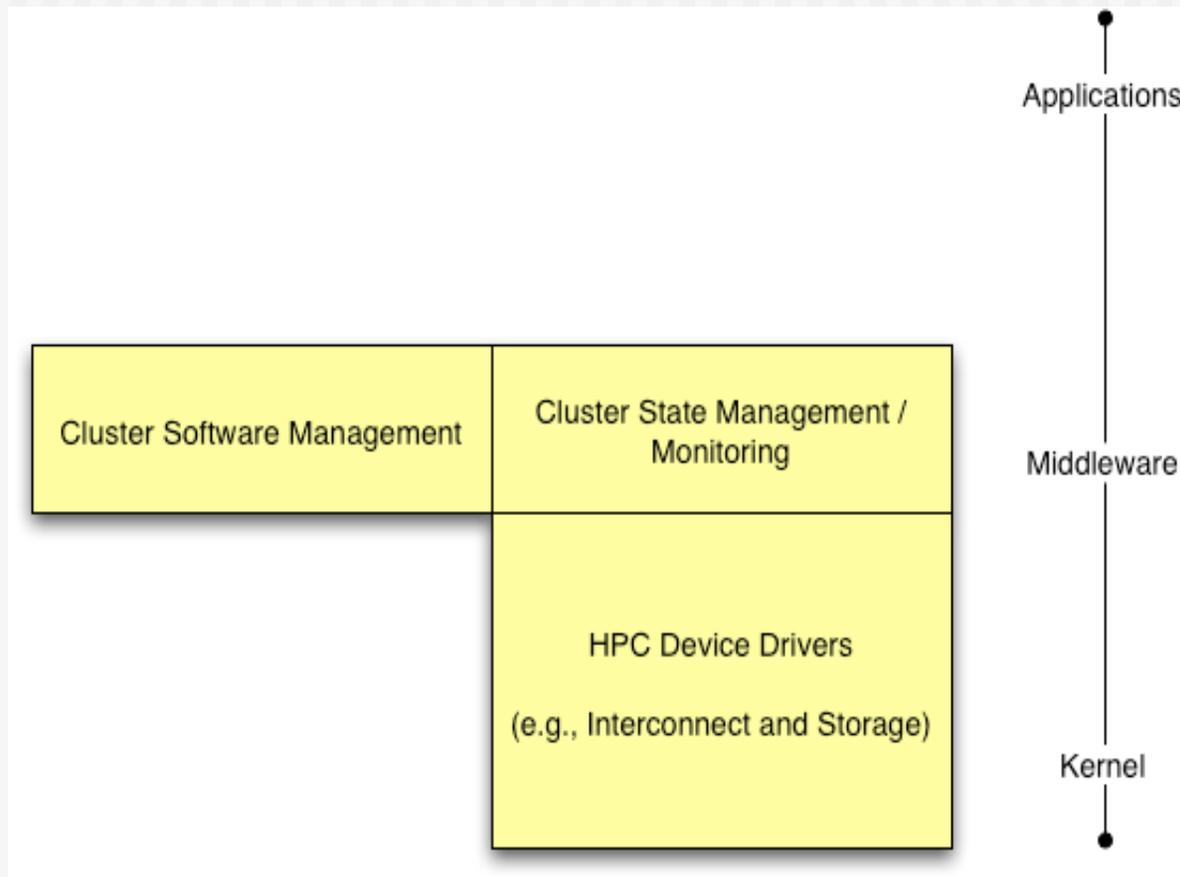
# Network File System

---

- ◆ User account are served over NFS
  - ⇒ Works for small clusters ( $\leq 128$  nodes)
  - ⇒ Will not work for large clusters ( $>1024$  nodes)
  - ⇒ Network Attached Storage (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



# Rocks Cluster Software





# 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
- ◆ Video
  - ⦿ nVidia (from Viz Roll)
- ◆ Add your own
  - ⦿ Cluster Gigabit Ethernet driver
  - ⦿ Infiniband driver
- ◆ Kernel Modules are dynamically built
- ◆ No need to manage binary Kernel Modules
- ◆ Burn CPU time, not human time



# 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

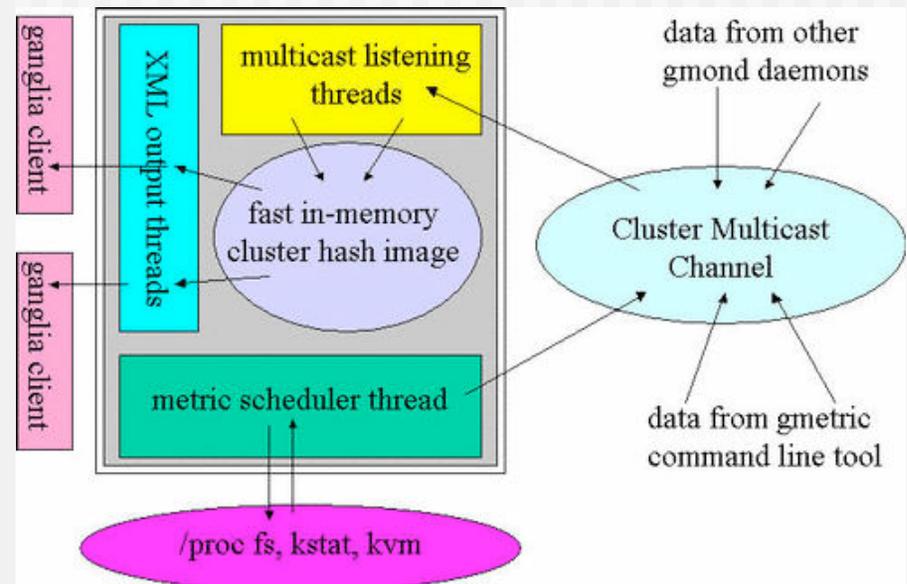
- ◆ Remotely Interact with Installation
  - Initial kickstart
  - Re-Installation
- ◆ Shoot-node
  - Reinstall OS and brings up eKV
- ◆ eKV
  - Ssh to node while it is installing
  - See the console output over Ethernet
- ◆ Newer versions of Rocks (4.0+) use VNC
  - Graphical
  - Works on headless machines





# Ganglia (or SCMSWeb / SCE Roll)

- ◆ Scalable cluster monitoring system
  - Based on IP multi-cast
  - Matt Massie, et al from UCB
  - <http://ganglia.sourceforge.net>
- ◆ Gmond 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 Screenshot



Host Report for Tue, 18 Mar 2003 01:28:58 +0000 [Get Fresh Data](#)

Last  [Node View](#) 

[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_idle	97.1 %
cpu_num	2
cpu_speed	900 MHz
mem_total	1011568 KB
mtu	1500 B
swap_total	1048544 KB

#### britannic LOAD last hour

Legend: 1-Minute Load (grey), Total CPUs (red), Running Processes (blue)

#### britannic CPU last hour

Legend: User CPU (blue), Nice CPU (yellow), System CPU (red), Idle CPU (grey)

#### britannic MEM last hour

Legend: Memory Used (blue), Memory Shared (dark blue), Memory Cached (green), Memory Buffered (light green), Memory Swapped (purple), Total In-Core Memory (red)



# SCMSWeb Screenshot



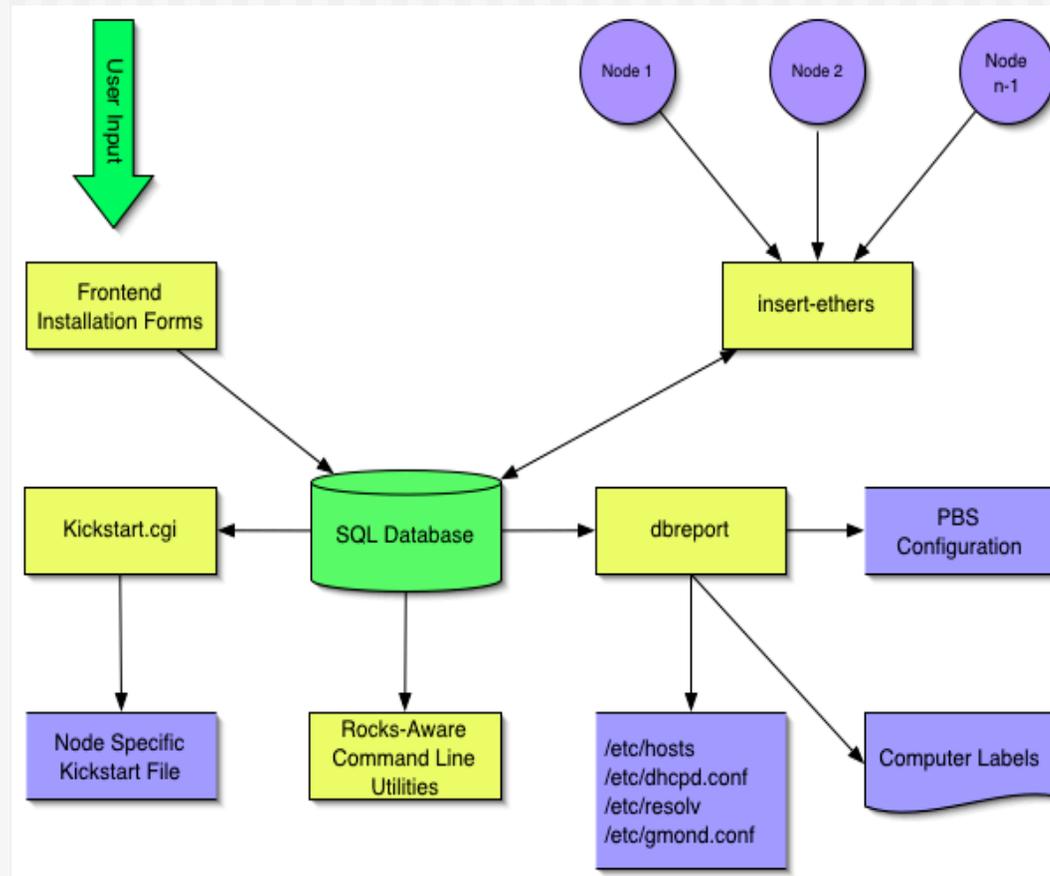
# Cluster State Management

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





# 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

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

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

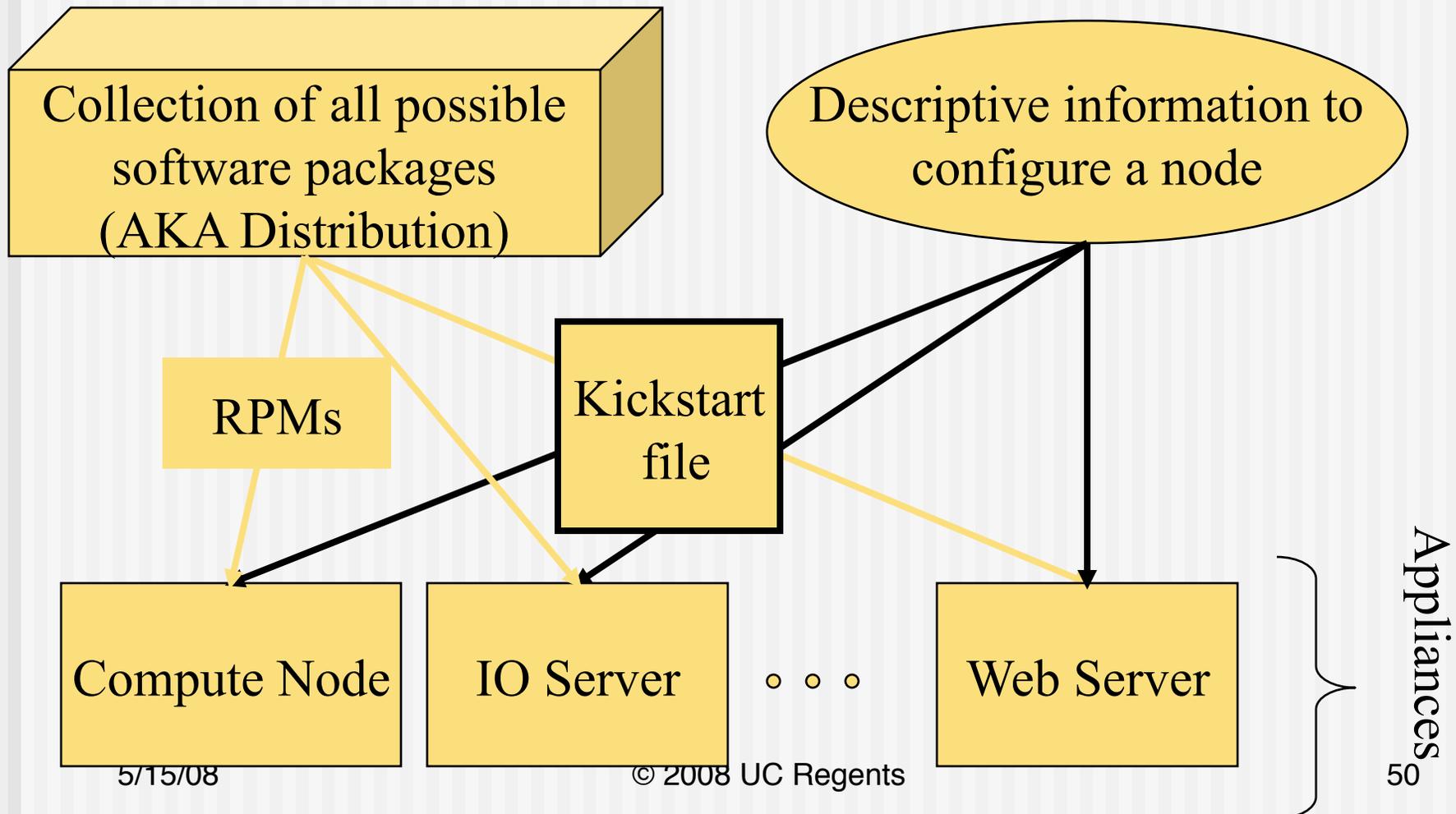
Distributions	
ID	Primary Key
Name	Distribution name
Release	Release Path
Lang	Release Language

- ⇒ E.g., get the hostname for the bottom 8 nodes of each cabinet:

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

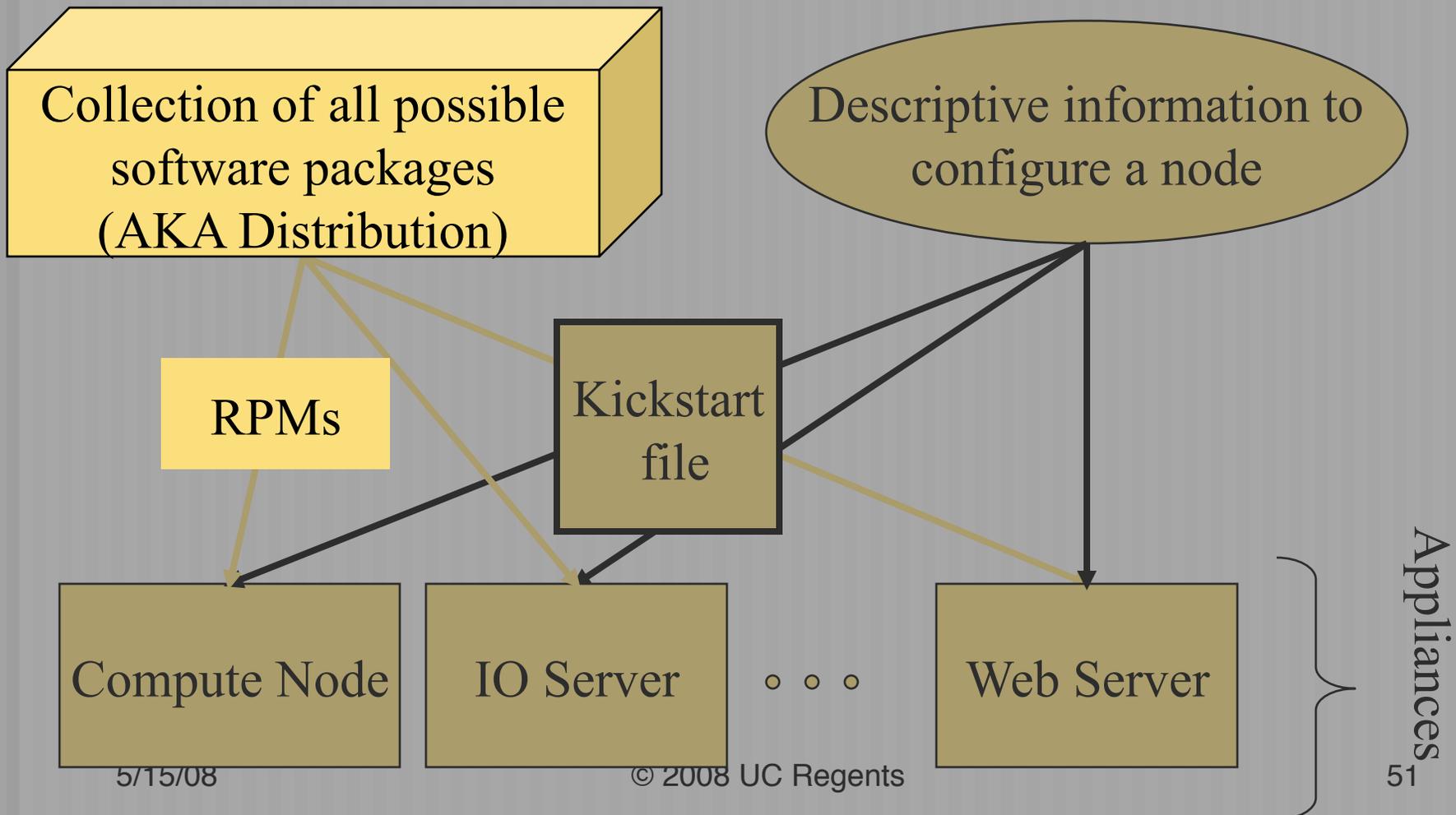


# Software Installation



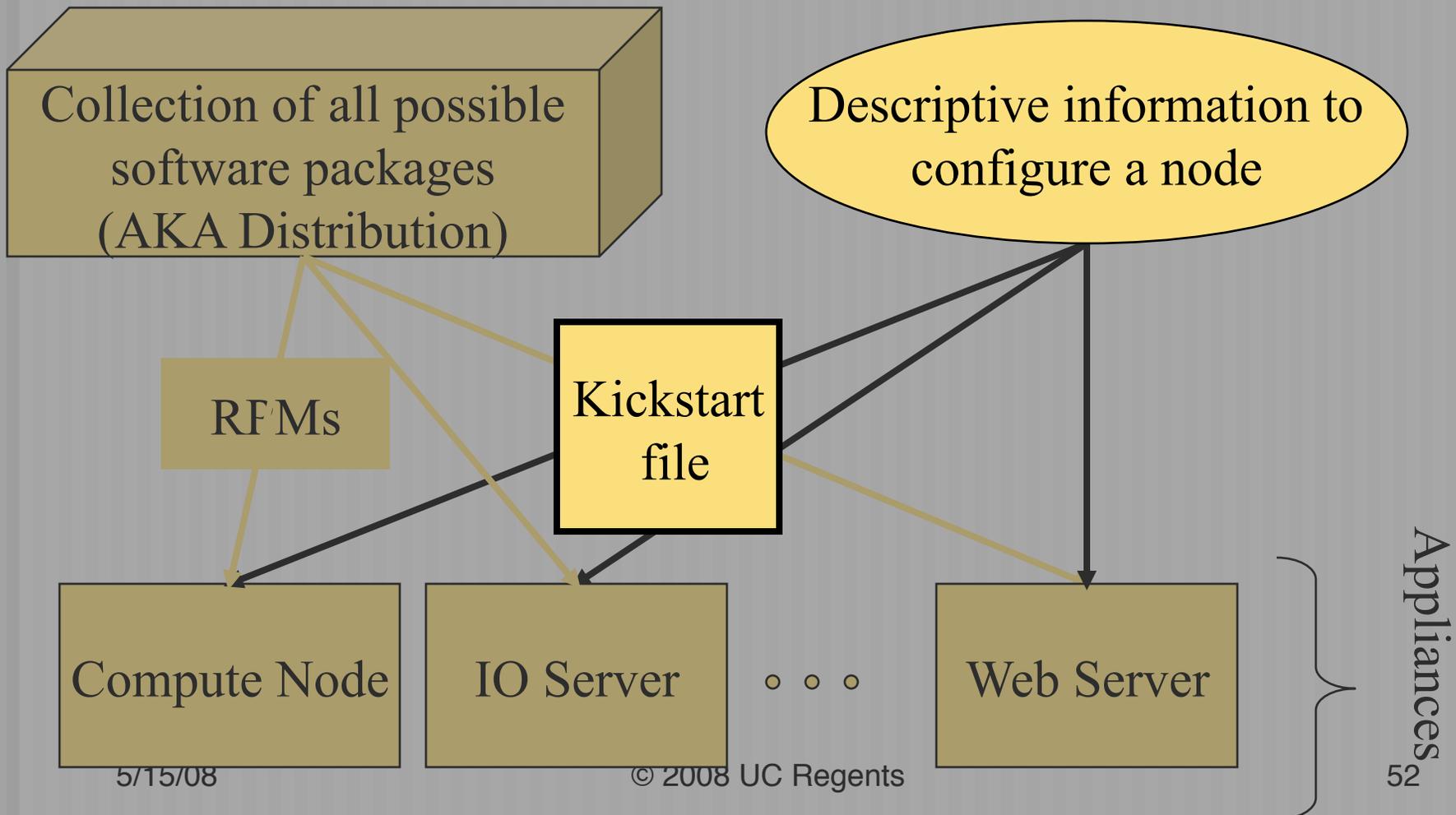


# Software Repository





# Installation Instructions





# Cluster Software Management

## Software Packages

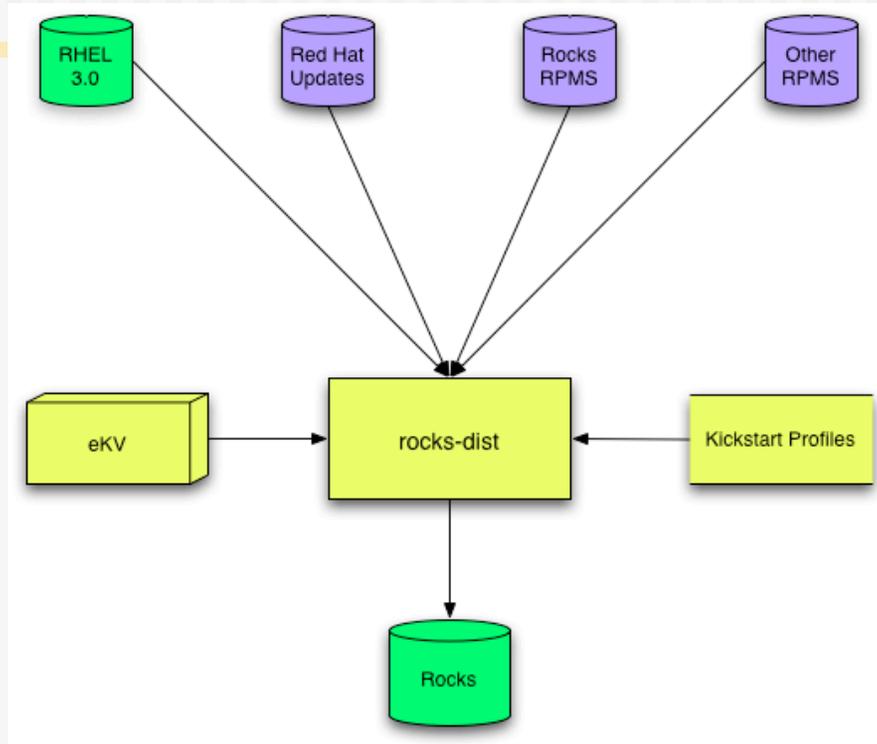
- ◆ RPMs
  - ➔ Standard Red Hat (desktop) packaged software
  - ➔ Or your own addons
- ◆ Rocks-dist
  - ➔ Manages the RPM repository
  - ➔ This is the distribution

## Software Configuration

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



# Building a Rocks Distribution

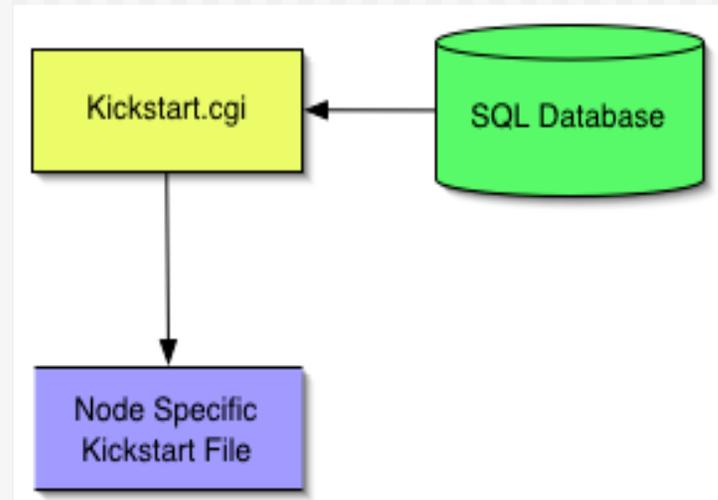


- ◆ Start with Red Hat
- ◆ Add updates, Rocks (and optional other) software
- ◆ Add Kickstart profiles
- ◆ Modify Red Hat installation boot image
- ◆ Resulting in a Red Hat compatible Rocks distribution



# 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



# What is a Kickstart File?

## ◆ Setup & Packages (20%)

```
cdrom
zerombr yes
bootloader --location mbr --useLilo
skipx
auth --useshadow --enablemd5
clearpart --all
part /boot --size 128
part swap --size 128
part / --size 4096
part /export --size 1 --grow
lang en_US
langsupport --default en_US
keyboard us
mouse genericps/2
timezone --utc GMT
rootpw --iscrypted nrDq4Vb42jjQ.
text
install
reboot

%packages
@Base
@Emacs
@GNOME
```

## ◆ Post Configuration (80%)

```
%post

cat > /etc/nsswitch.conf << 'EOF'
passwd:    files
shadow:    files
group:     files
hosts:     files dns
bootparams: files
ethers:    files
EOF

cat > /etc/ntp.conf << 'EOF'
server ntp.ucsd.edu
server      127.127.1.1
fudge      127.127.1.1 stratum 10
authenticate no
driftfile /etc/ntp/drift
EOF

/bin/mkdir -p /etc/ntp
cat > /etc/ntp/step-tickers << 'EOF'
ntp.ucsd.edu
EOF

/usr/sbin/ntpdate ntp.ucsd.edu
/sbin/hwclock --systohc
```



# Issues

---

- ◆ High level description of software installation
  - List of packages (RPMs)
  - System configuration (network, disk, accounts, ...)
  - Post installation scripts
- ◆ *De facto* standard for Linux
- ◆ Single ASCII file
  - Simple, clean, and portable
  - Installer can handle simple hardware differences
- ◆ Monolithic
  - No macro language
  - Differences require forking (and code replication)
  - Cut-and-Paste is not a code re-use model

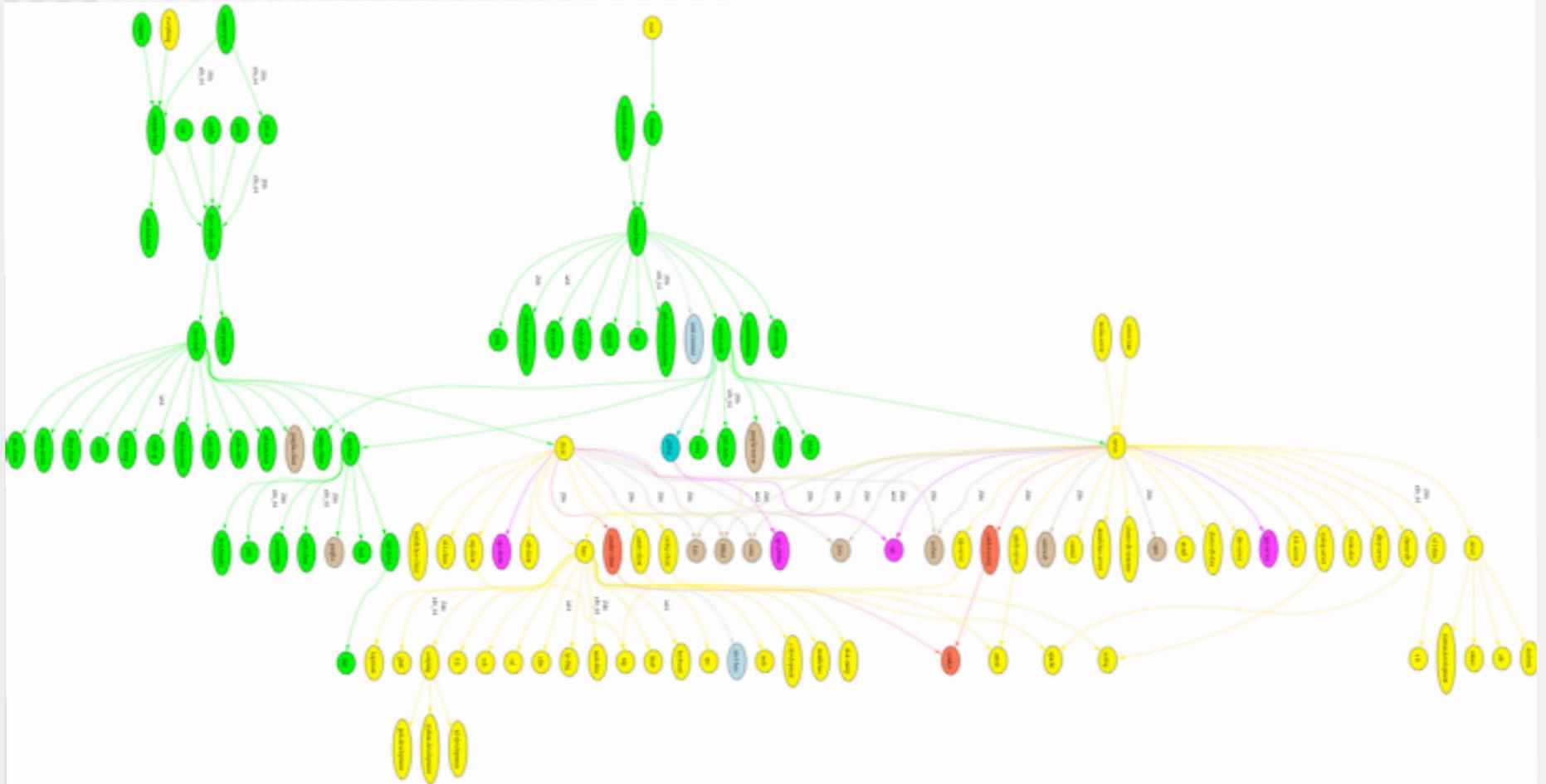


---

# **XML KICKSTART**



# It looks something like this





# Implementation

---

## ◆ Nodes

- Single purpose modules
- Kickstart file snippets (XML tags map to kickstart commands)
- Approximately 200 node files in Rocks

## ◆ Graph

- Defines interconnections for nodes
- Think OOP or dependencies (class, #include)
- A single default graph file in Rocks

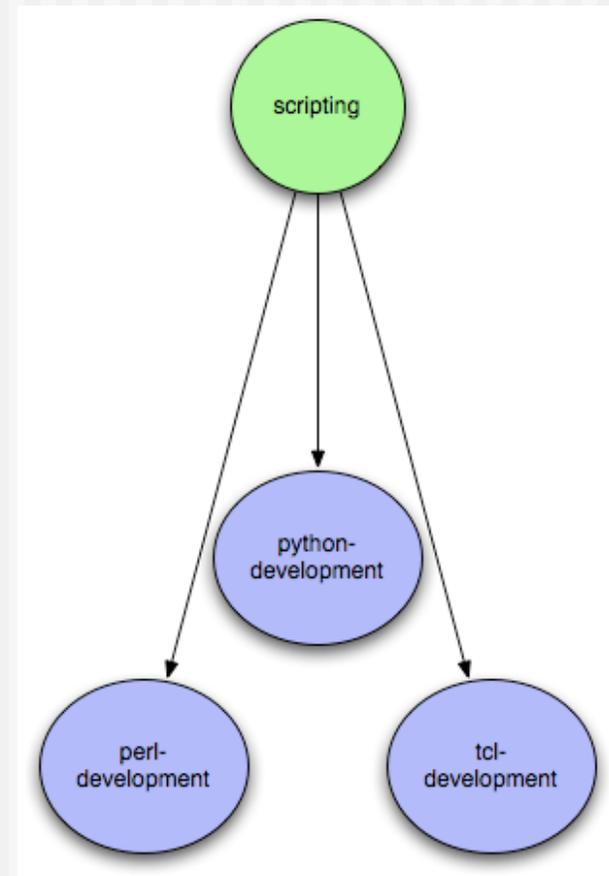
## ◆ Macros

- SQL Database holds site and node specific state
- Node files may contain `<var name="state"/>` tags



# Composition

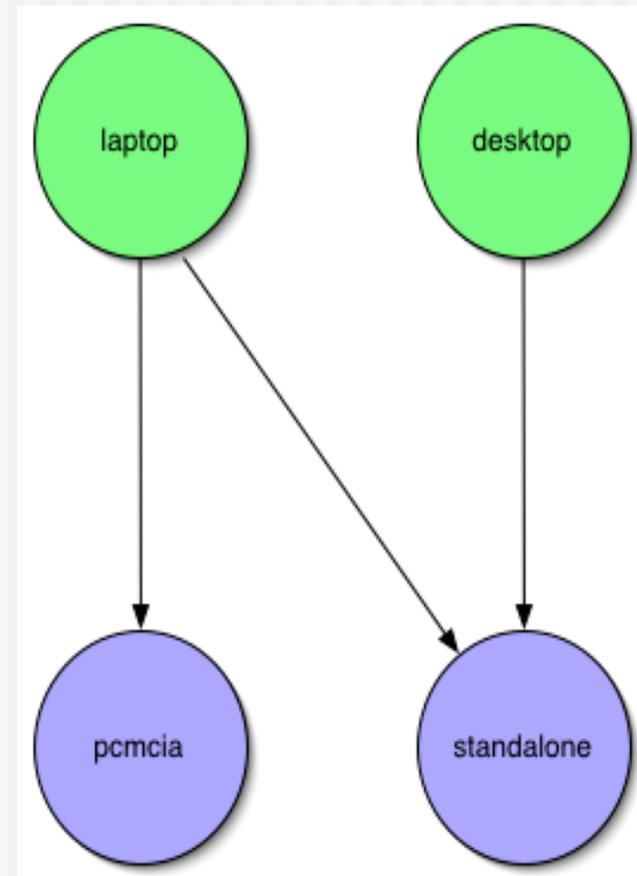
- ◆ Aggregate Functionality
- ◆ Scripting
  - ⇒ IsA perl-development
  - ⇒ IsA python-development
  - ⇒ IsA tcl-development





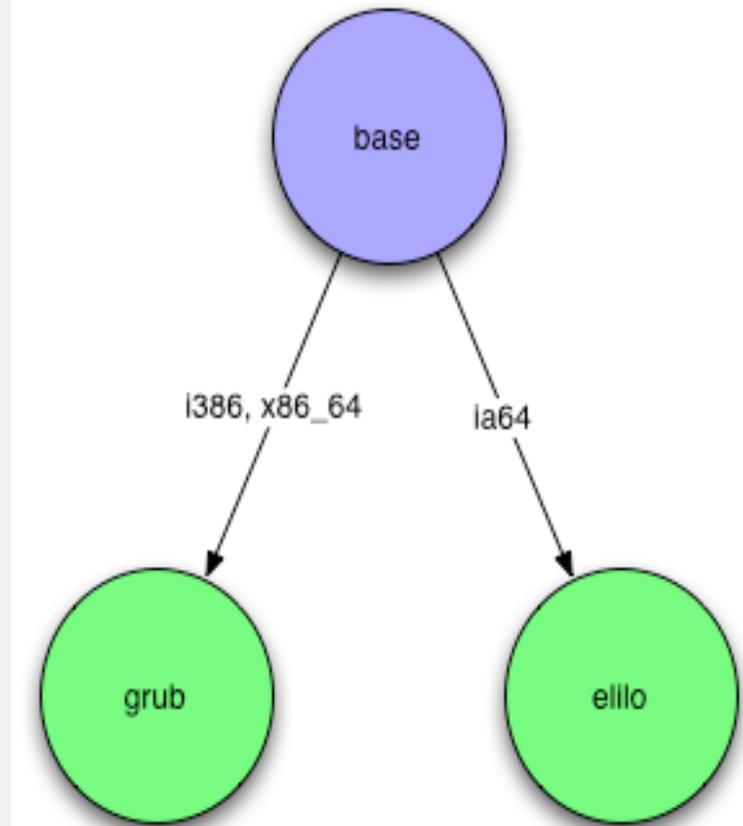
# Appliances

- ◆ Laptop / Desktop
  - ⇒ Appliances
  - ⇒ Final classes
  - ⇒ Node types
- ◆ Desktop IsA
  - ⇒ standalone
- ◆ Laptop IsA
  - ⇒ standalone
  - ⇒ Pcmcia
- ◆ Specify only the differences



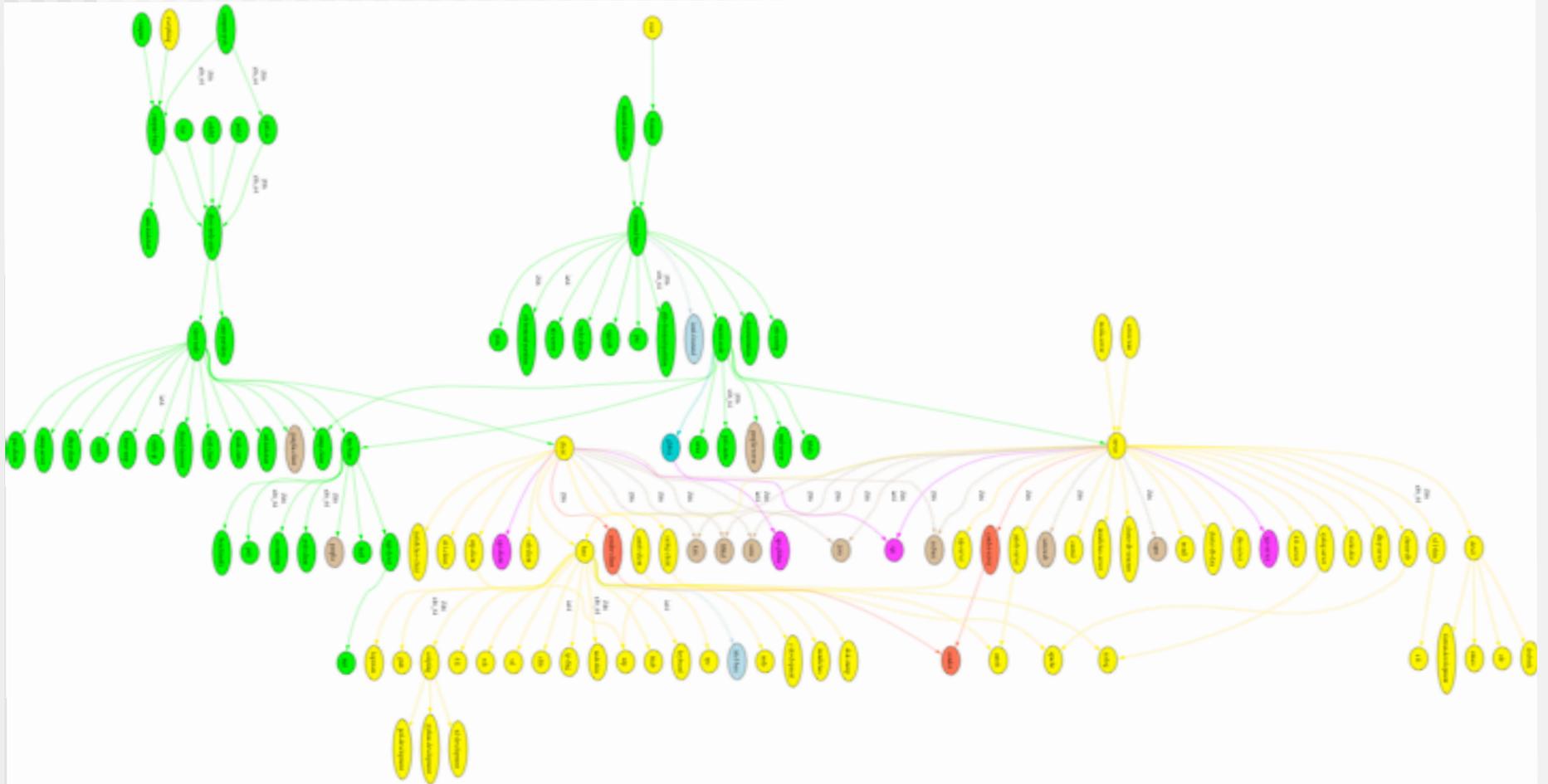
# Architecture Differences

- ◆ Conditional inheritance
- ◆ Annotate edges with target architectures
- ◆ if i386
  - Base ISA grub
- ◆ if ia64
  - Base ISA elilo
- ◆ One Graph, Many CPU Architectures
  - Heterogeneity becomes easy
  - Not true for SSI or Imaging





# Putting in all together





# 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>
```



---

# CLUSTER SQL DATABASE

# Cluster State Management

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





# Nodes and Groups

ID	MAC	Name	Membership	Hardware	Rack	Rank	IP	C
1		frontend-0	1	0	0	0	10.1.1.1	
2	00:30:c1:d8:59:00	network-1-0	6	0	1	0	10.255.255.254	
3	00:01:e7:1a:be:00	network-0-0	6	0	0	0	10.255.255.253	
4	00:30:c1:d8:ac:80	network-3-0	6					
5	00:50:8b:a5:4d:b1	nfs-0-0	12					
6	00:50:8b:c5:c3:72	nfs-0-1	12					
7	00:50:8b:a5:57:ff	nfs-1-0	12					
8	00:50:8b:a5:4c:f4	nfs-1-1	12					
9	00:50:8b:e0:3a:a7	compute-0-0	2					
10	00:50:8b:e0:44:5e	compute-0-1	2					
11	00:50:8b:e0:40:95	compute-0-2	2					
12	00:50:8b:e0:40:93	compute-0-3	2					
13	00:50:8b:e0:42:df	compute-0-4	2					

Nodes Table

ID	Name	Appliance	Distribution	Compute
1	Frontend	1	1	no
2	Compute	2	1	yes
3	PVFS I/O Node	3	1	no
4	Compute with PVFS	4	1	yes
5	Laptop	5	1	no
6	Ethernet Switches	6		no
7	Myrinet Switches	6		no
8	Power Units	7		no
9	Remote Management	8		no
10	DTF Compute	9	1	yes
11	Web Portal	10	1	no
12	NFS Server	11	1	no

Memberships Table



# Groups and Appliances

ID	Name	Appliance	Distribution	Compute
1	Frontend	1	1	no
2	Compute	2		yes
3	PVFS I/O Node	3		no
4	Compute with PVFS	4		yes
5	Laptop	5		no
6	Ethernet Switches	6		no
7	Myrinet Switches	6		no
8	Power Units	7		no
9	Remote Management	8		no
10	DTF Compute	9		yes
11	Web Portal	10		no
12	NFS Server	11	1	no

Memberships Table

ID	Name	ShortName	Graph	Node
1	frontend	f	default	frontend
2	compute	c	default	compute
3	pvfs	pv	default	pvfs-io
4	comp-pvfs	cp	default	compute-pvfs
5	laptop		default	laptop
6	network	n		
7	power	p		
8	manager			
9	dtf	d	default	dtf-compute
10	portal	pl	default	portal
11	nfs	n	default	nfs

Appliances Table



# Simple key - value pairs

ID	Membership	Service	Component	Value
1	0	Kickstart	PublicNTPHost	ntp.ucsd.edu
2	0	Kickstart	ZeroMBR	yes
3	0	Kickstart	PrivateKickstartCGI	kickstart.cgi
4	0	Kickstart	PublicNetmask	255.255.255.0
5	0	Kickstart	PublicNetwork	192.31.21.0
6	0	Kickstart	PrivateNISMaster	frontend-0
7	0	Kickstart	PrivateHostname	frontend-0
8	0	Kickstart	PrivateIPForwarding	yes
9	0	Kickstart	PrivateGateway	10.1.1.1
10	0	Kickstart	PublicKickstartBasedir	install
11	0	Kickstart	Lang	en_US

- ◆ Used to configure DHCP and to customize appliance kickstart files



# Nodes XML Tools: `<var>`

## ◆ Get Variables from Database

- `<var name="Kickstart_PrivateGateway" />`
- `<var name="Node_Hostname" />`

```
10.1.1.1  
compute-0-0
```

- Can grab any value from the *app\_globals* database table



# Nodes XML Tools: `<eval>`

- ◆ Do processing on the frontend:
  - `<eval shell="bash">`
- ◆ To insert a fortune in the kickstart file:

```
<eval shell="bash">  
/usr/games/fortune  
</eval>
```

```
"Been through Hell?  
Whaddya bring back for  
me?"  
-- A. Brilliant
```



# Nodes XML Tools <file>

- ◆ Create a file on the system:  
`<file name="/etc/hi-mom" mode="append">`  
    How are you today?  
`</file>`
- ◆ Used extensively throughout Rocks post sections
  - Keeps track of alterations automatically via RCS.

```
<file name="/etc/hi" perms="444">  
How are you today?  
I am fine.  
</file>
```

```
...RCS checkin commands...  
cat > /etc/hi << 'EOF'  
How are you today?  
I am fine.  
EOF  
chmod 444 /etc/hi-mom  
...RCS cleanup commands...
```



# Fancy <file>: nested tags

```
<file name="/etc/hi">
```

Here is your fortune for today:

```
<eval>
```

```
date +"%d-%b-%Y"
```

```
echo ""
```

```
/usr/games/fortune
```

```
</eval>
```

```
</file>
```

...RCS checkin commands...

```
cat > /etc/hi << 'EOF'
```

**Here is your fortune for today:**

**13-May-2005**

**"Been through Hell? Whaddya  
bring back for me?"**

**-- A. Brilliant**

**EOF**

...RCS cleanup commands...



# Nodes Main

- ◆ Used to specify basic configuration:
  - timezone
  - mouse, keyboard types
  - install language
- ◆ Used more rarely than other tags
- ◆ Rocks main tags are usually a straight translation:

```
<main>  
  
  <timezone>America/Mission_Beach  
  </timezone>  
  
</main>
```

```
...  
timezone America/Mission_Beach  
...  
rootpw --iscrypted sndk48shdlwis  
mouse genericps/2  
url --url http://10.1.1.1/install/rocks-dist/..
```



# Nodes Packages

- ◆ `<package>java</package>`
  - Specifies an RPM package. Version is automatically determined: take the *newest* rpm on the system with the name 'java'.
- ◆ `<package arch="x86_64">java</package>`
  - Only install this package on x86\_64 architectures
- ◆ `<package arch="i386,x86_64">java</package>`

```
<package>newcastle</package>  
<package>stone-pale</package>  
<package>guinness</package>
```

```
%packages  
newcastle  
stone-pale  
guinness
```



# Nodes Packages

- ◆ RPM name is a basename (not fullname of RPM)
  - ➔ For example, RPM name of package below is 'kernel'

```
# rpm -qip /home/install/rocks-dist/lan/i386/RedHat/RPMS/kernel-2.6.9-22.EL.i686.rpm
Name       : kernel                Relocations: (not relocatable)
Version    : 2.6.9                 Vendor: CentOS
Release    : 22.EL                Build Date: Sun 09 Oct 2005 03:01:51 AM WET
Install Date: (not installed)     Build Host: louisahome.local
Group      : System Environment/Kernel  Source RPM: kernel-2.6.9-22.EL.src.rpm
Size       : 25589794             License: GPLv2
Signature  : DSA/SHA1, Sun 09 Oct 2005 10:44:40 AM WET, Key ID a53d0bab443e1821
Packager   : Johnny Hughes <johnny@centos.org>
Summary    : the linux kernel (the core of the linux operating system)
Description:
The kernel package contains the Linux kernel (vmlinuz), the core of any
Linux operating system
```



# Nodes Post

## ntp-client.xml

```
<post>
```

```
/bin/mkdir -p /etc/ntp
```

```
/usr/sbin/ntpdate <var name="Kickstart_PrivateNTPHost"/>
```

```
/sbin/hwclock --systohc
```

```
</post>
```

```
%post
```

```
/bin/mkdir -p /etc/ntp
```

```
/usr/sbin/ntpdate 10.1.1.1
```

```
/sbin/hwclock --systohc
```

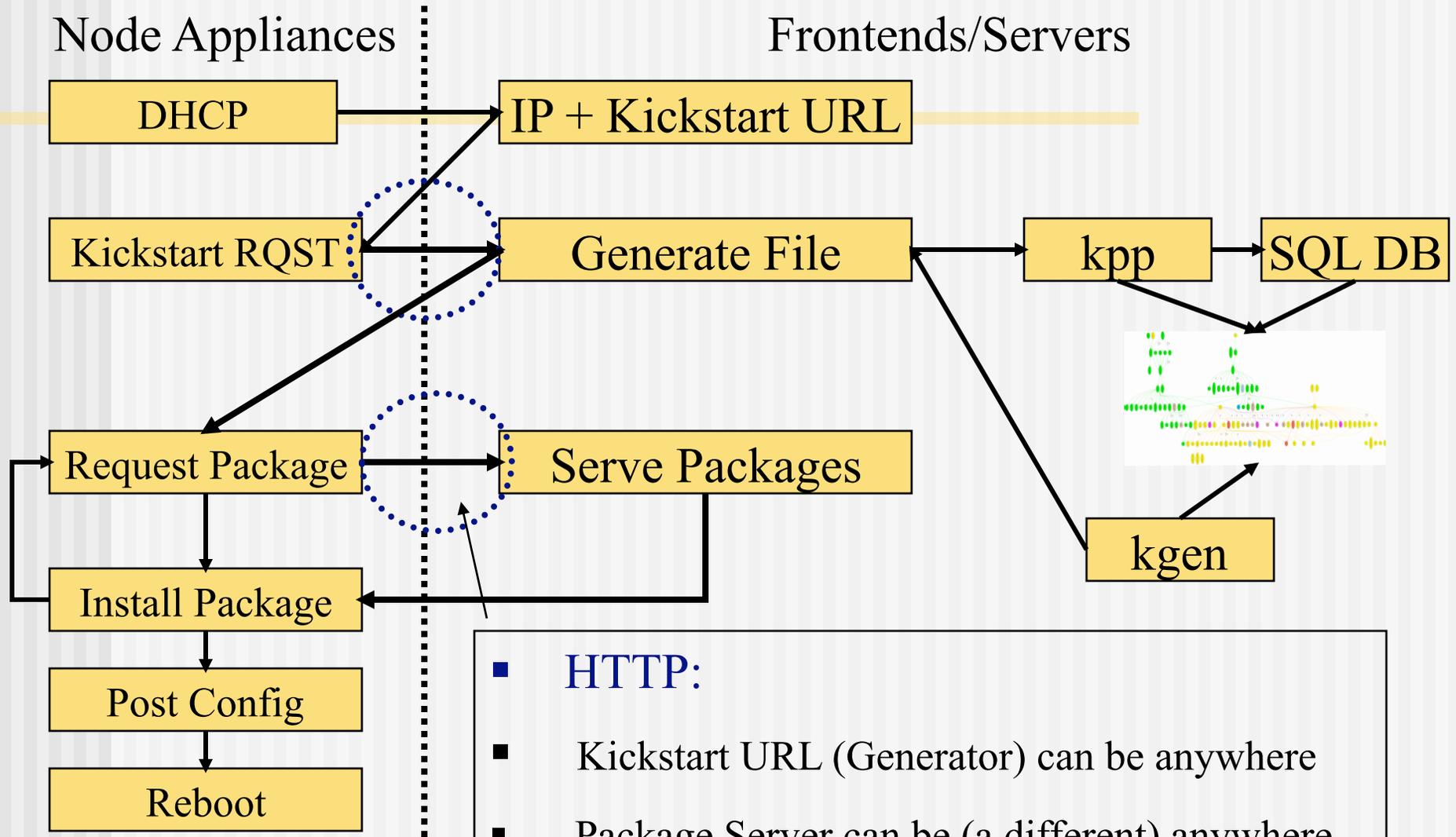


---

# PUTTING IT TOGETHER



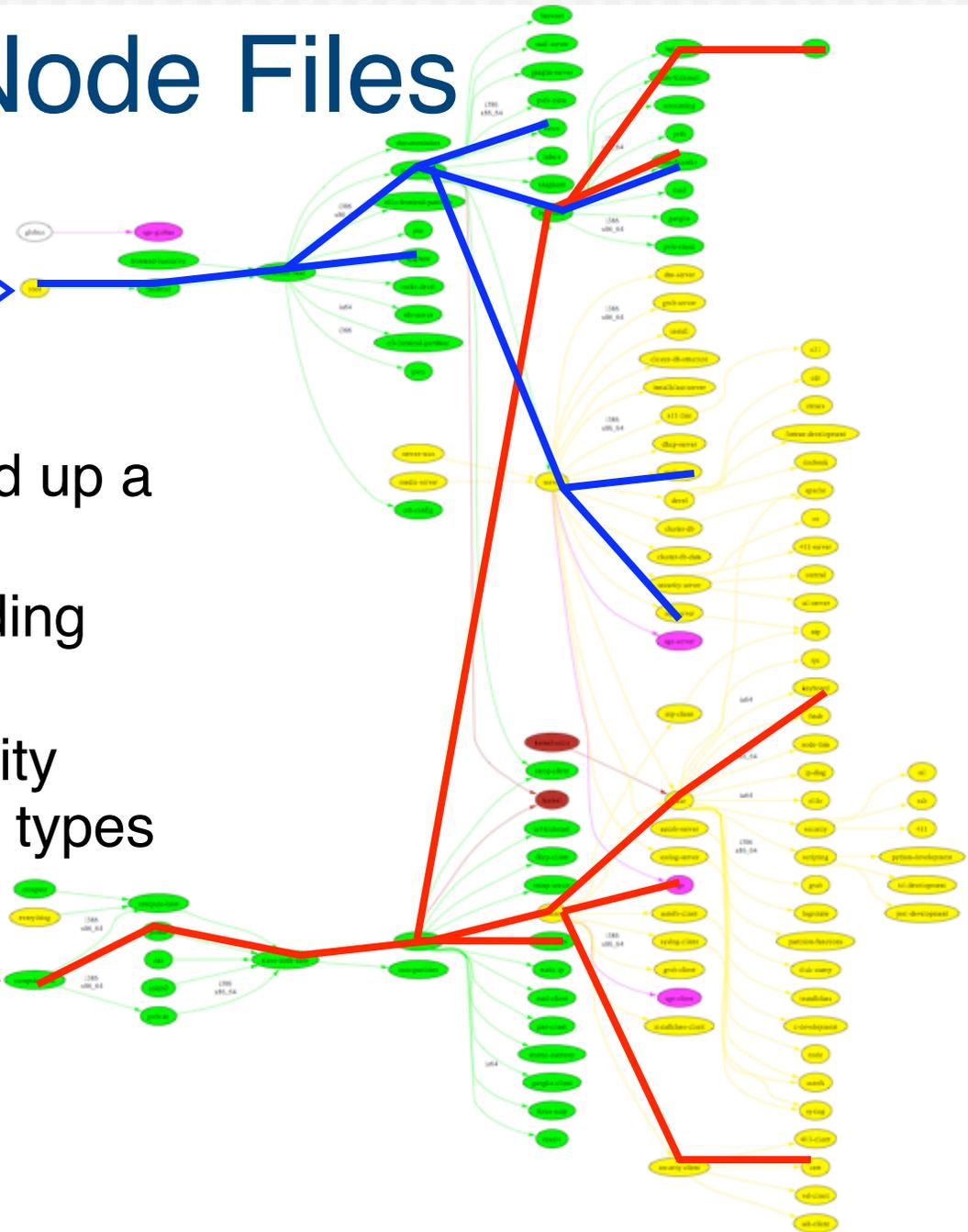
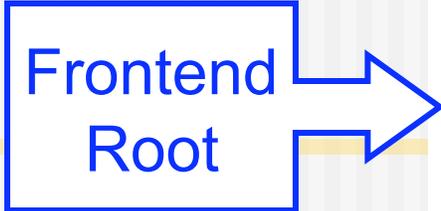
# Space-Time and HTTP



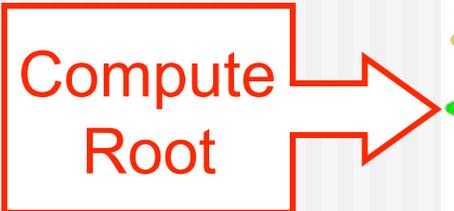
- HTTP:
- Kickstart URL (Generator) can be anywhere
- Package Server can be (a different) anywhere



# Gathering Node Files



- ◆ Traverse a graph to build up a kickstart file
- ◆ Makes kickstart file building flexible
- ◆ Easy to share functionality between disparate node types





# Another Look at XML

```
<graph>
```

```
  <edge from="client">
```

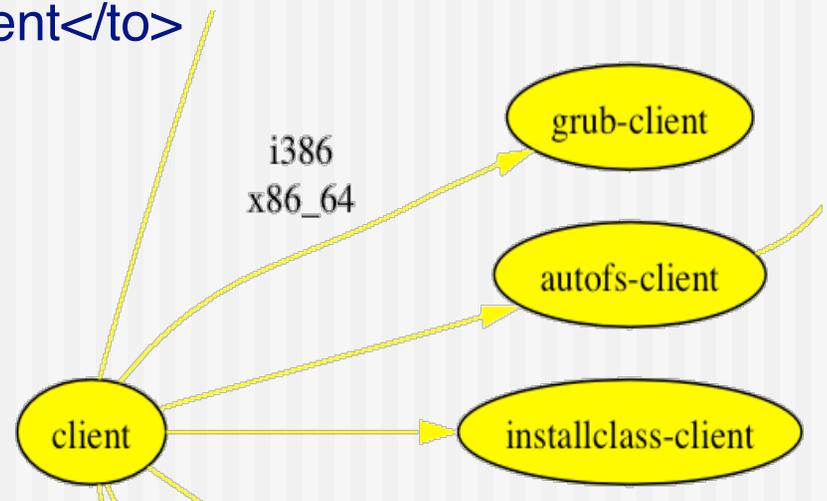
```
    <to arch="i386,x86_64">grub-client</to>
```

```
    <to>autofs-client</to>
```

```
    <to>installclass-client</to>
```

```
  </edge>
```

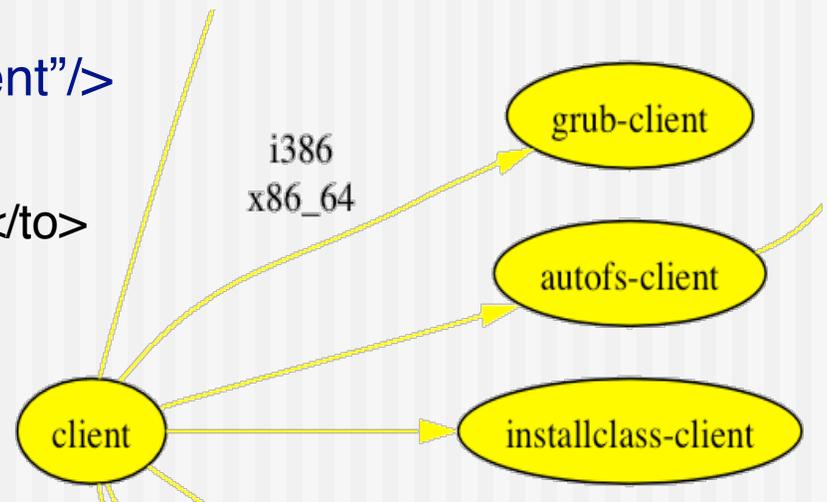
```
</graph>
```





# Partial Ordering

```
<graph>  
  <order head="autofs-client" tail="client"/>  
  <edge from="client">  
    <to arch="i386,x86_64">grub-client</to>  
    <to>autofs-client</to>  
    <to>installclass-client</to>  
  </edge>  
</graph>
```

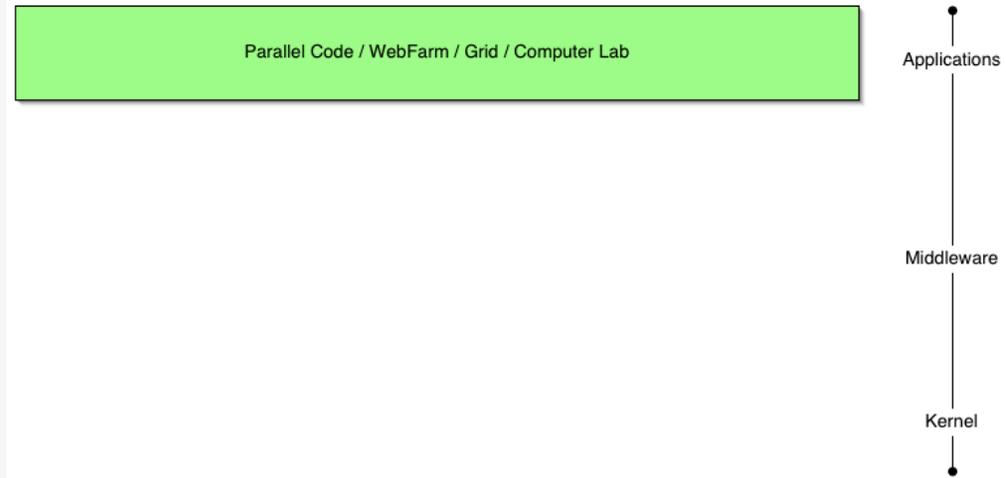


- ◆ Forces autofs-client <post> section to run before client's <post> section
- ◆ In order graph traversal enforces a partial ordering
- ◆ Applying standard graph theory to system installation



# Application Layer

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





---

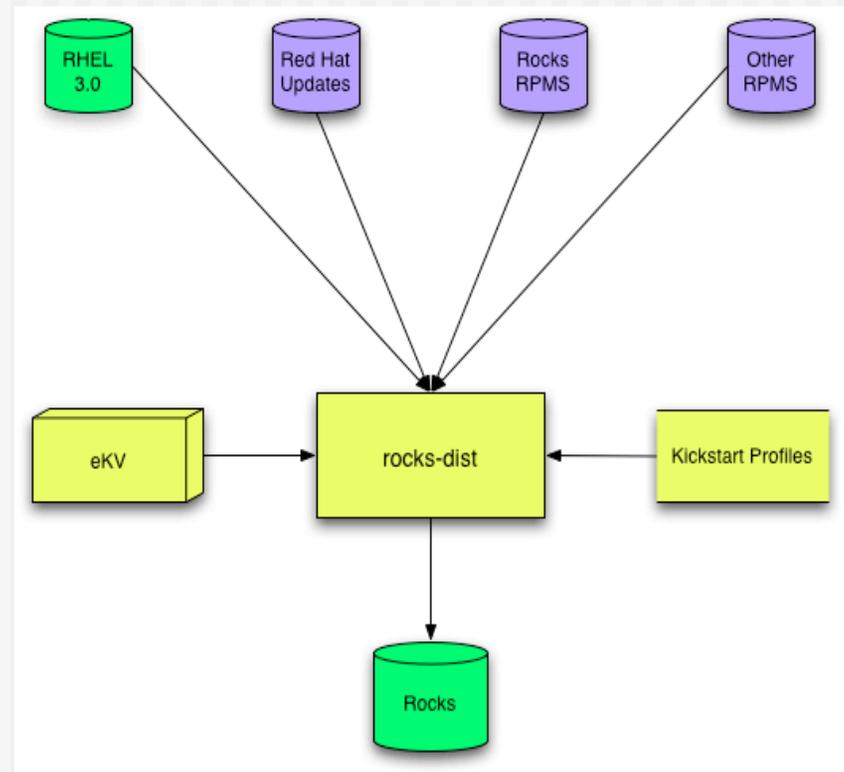
Inheritance and Rolls

# **BUILDING ON TOP OF ROCKS**



# How Rocks is Built

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





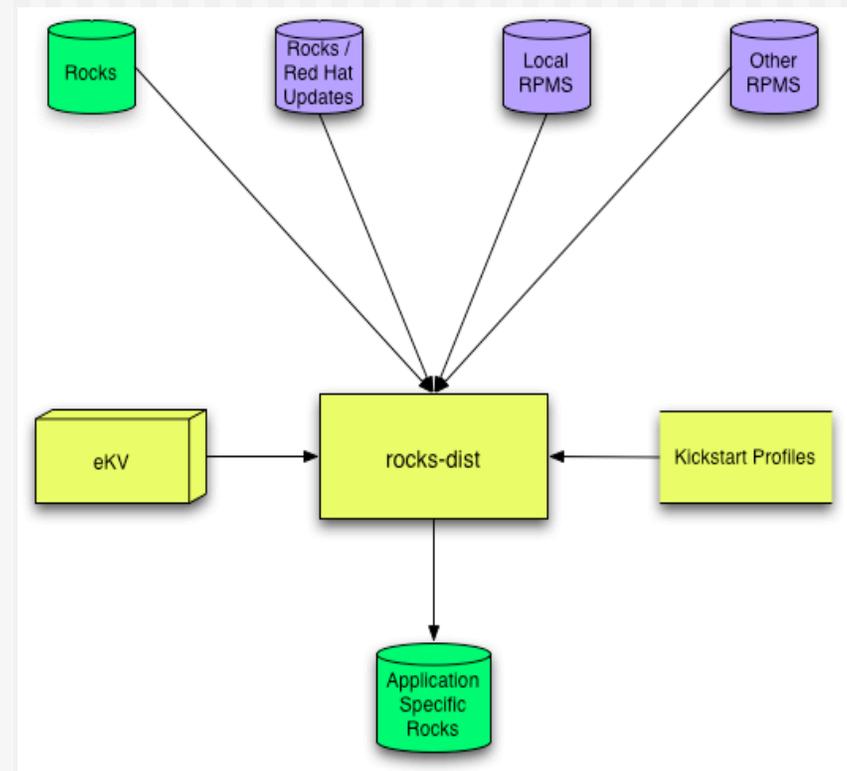
# How You Create Your Own Rocks

## ◆ Rocks-dist

- ⇒ Merges all RPMs
  - Rocks
  - Yours
- ⇒ Resolves versions
- ⇒ Creates Rocks++

## ◆ Your distribution

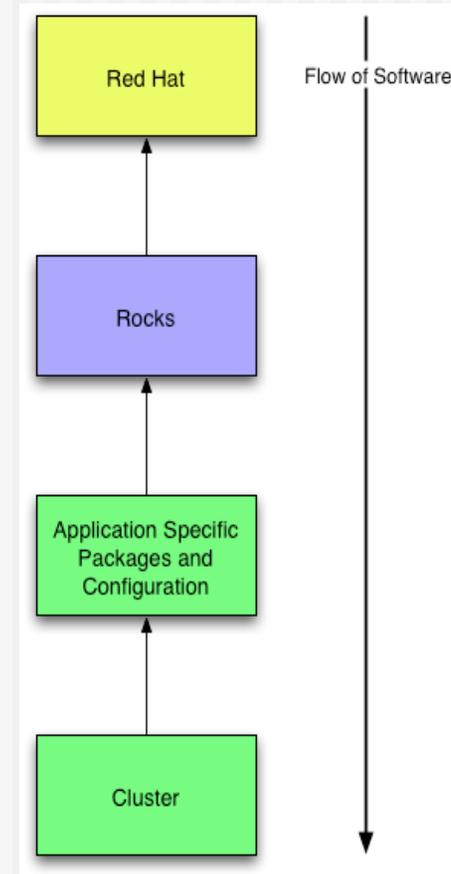
- ⇒ Looks just like Rocks
- ⇒ Application optimized Rocks





# 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?





# Need Better Flexibility in Stack

## ◆ Issues

- ➔ Static Stack
  - Cannot redefine
  - Cannot extend
- ➔ Monolithic Stack
  - Cannot “opt out”
  - All or nothing solution
  - E.g. PBS not SGE

## ◆ What we need

- ➔ Dynamic Stack
- ➔ Component Based Stack
- ➔ User / Developer Extensible

PICK PACKAGES

- > COMBO #1: PREMIUM
- > COMBO #2: SPORT
- > COMBO #3: COLD WEATHER
- > NEXT STEP

MINI COOPER S

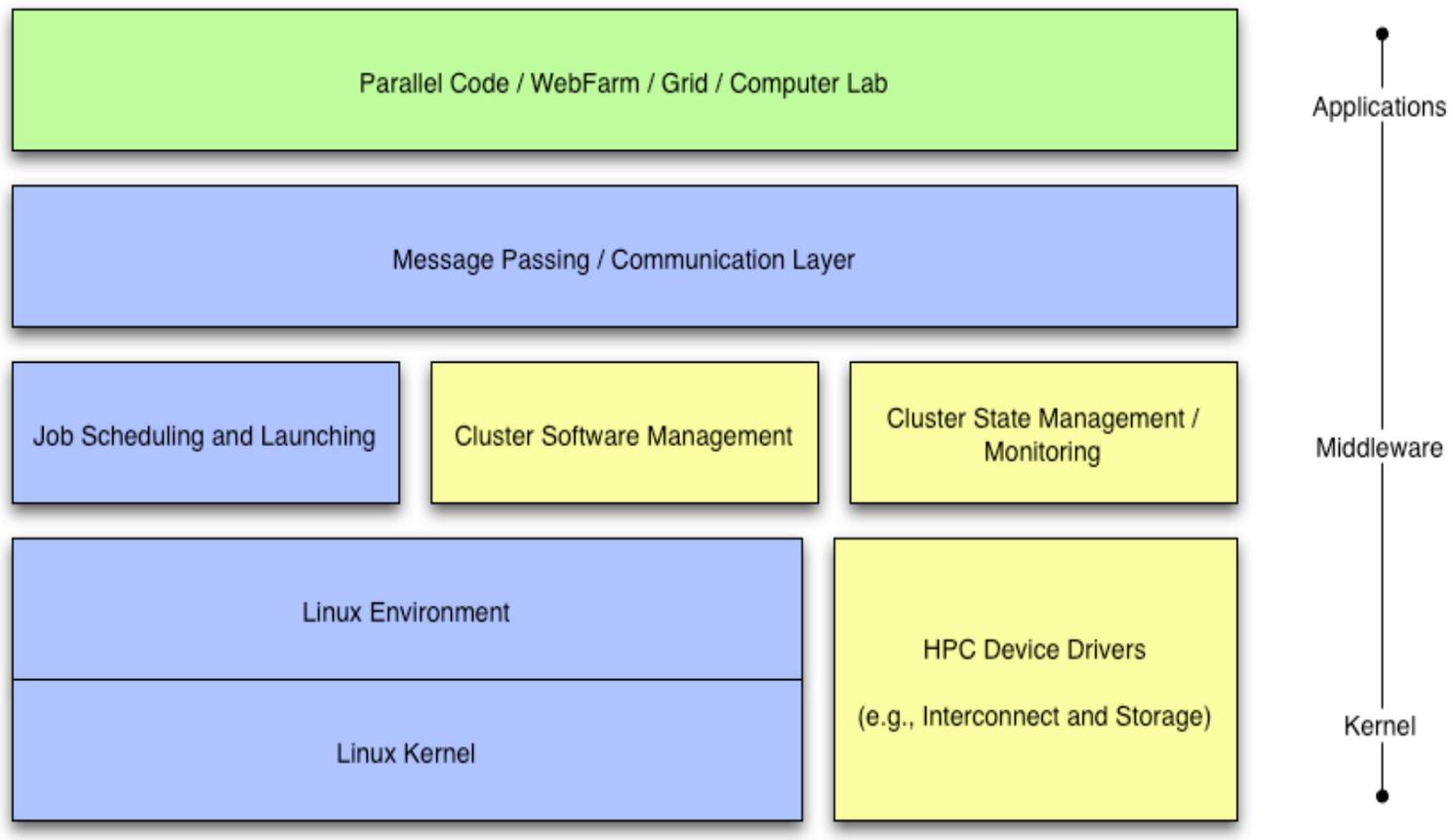
CLICK IMAGE TO ADD THE SPORT PACKAGE TO YOUR LIST.

THE SPORT PACKAGE WILL ADD:  
Dynamic stability control (DSC), bonnet stripes, xenon headlamps with powerwashers, front fog lamps, 17-inch alloy 5-lite wheels with 205/45 R17 performance or all-season run-flat tires.

Sport Package (\$1350)



# Rolls Break Apart Rocks



**Rolls: Modifying a Standard System Installer to Support User-Customizable Cluster Frontend Appliances.** Greg Bruno, Mason J. Katz, Federico D. Sacerdoti, and Phil M. Papadopoulos. *IEEE International Conference on Cluster Computing*, San Diego, California, Sep. 2004.



# 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

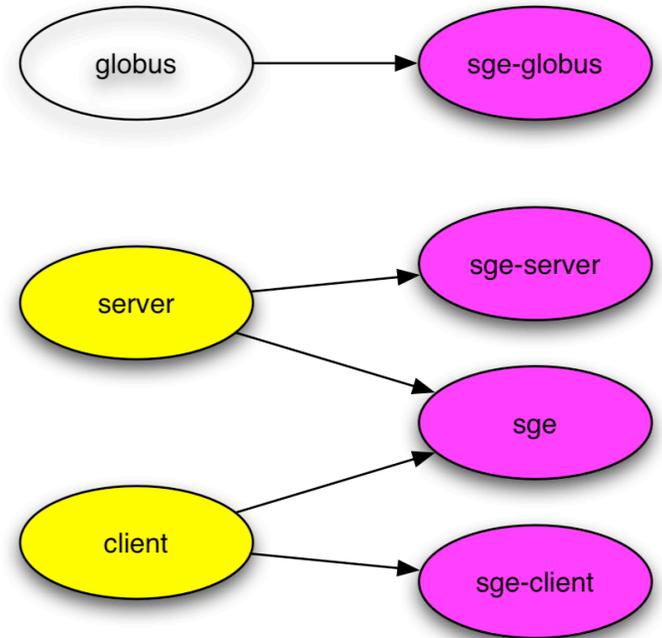
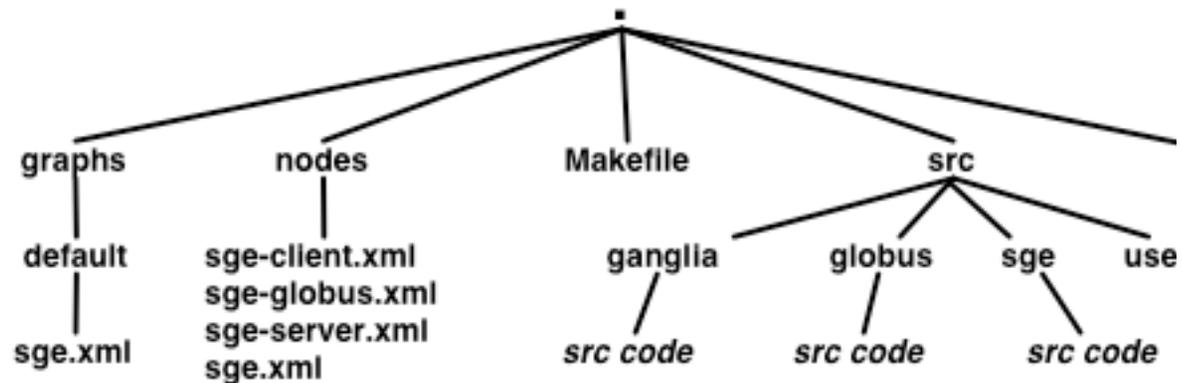
- ⊖ The core is basically stable and needs continued support
- ⊖ Rolls allow us to develop new ideas
- ⊖ Application Domain specific

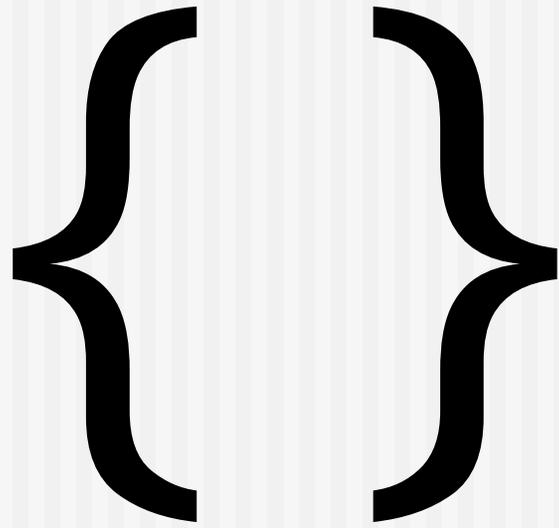
## ◆ IEEE Cluster 2004 - “Rolls: Modifying a Standard System Installer to Support User-Customizable Cluster Frontend Appliances”



# Rolls are sub-graphs

- ◆ A graph makes it easy to ‘splice’ in new nodes
- ◆ Each Roll contains its own nodes and splices them into the system graph file



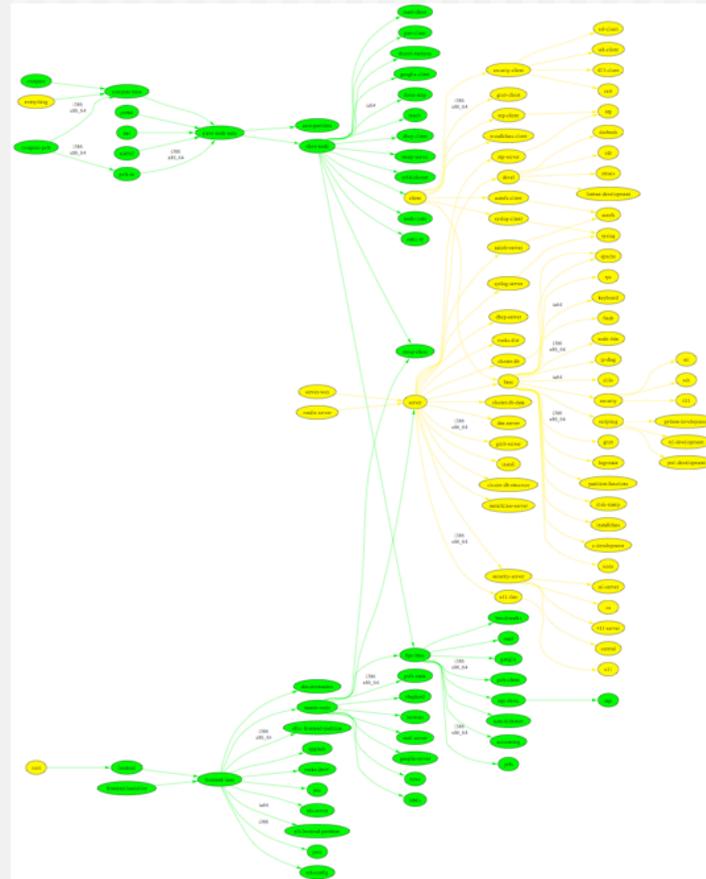


# STARTING FROM THE EMPTY SET



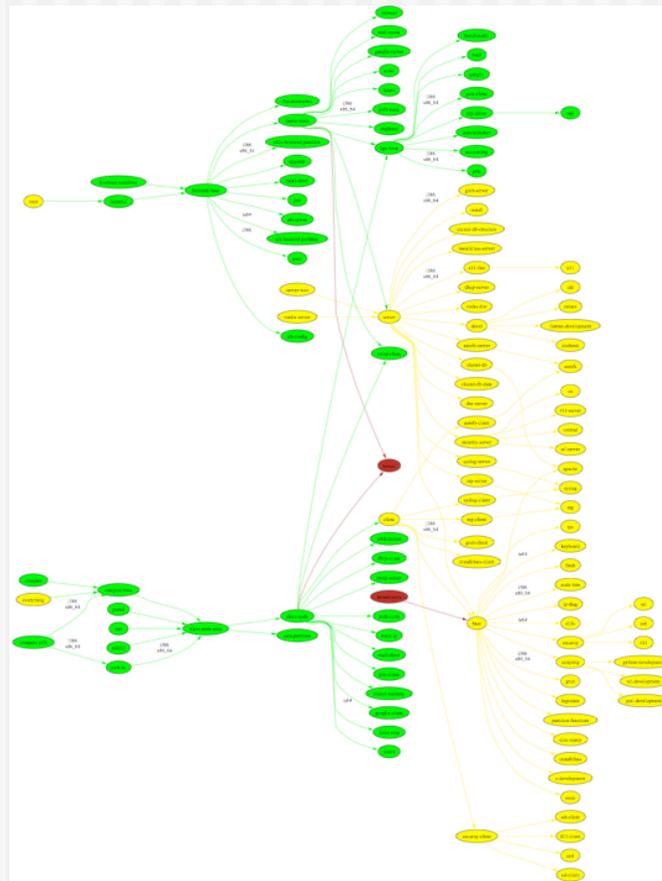


{ base, hpc }



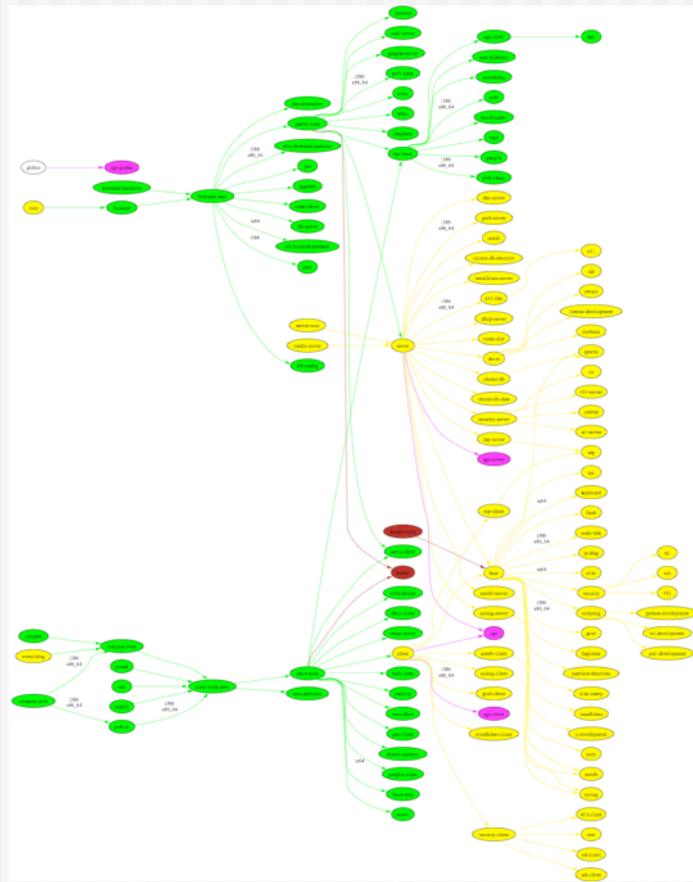


{ base, hpc, kernel }





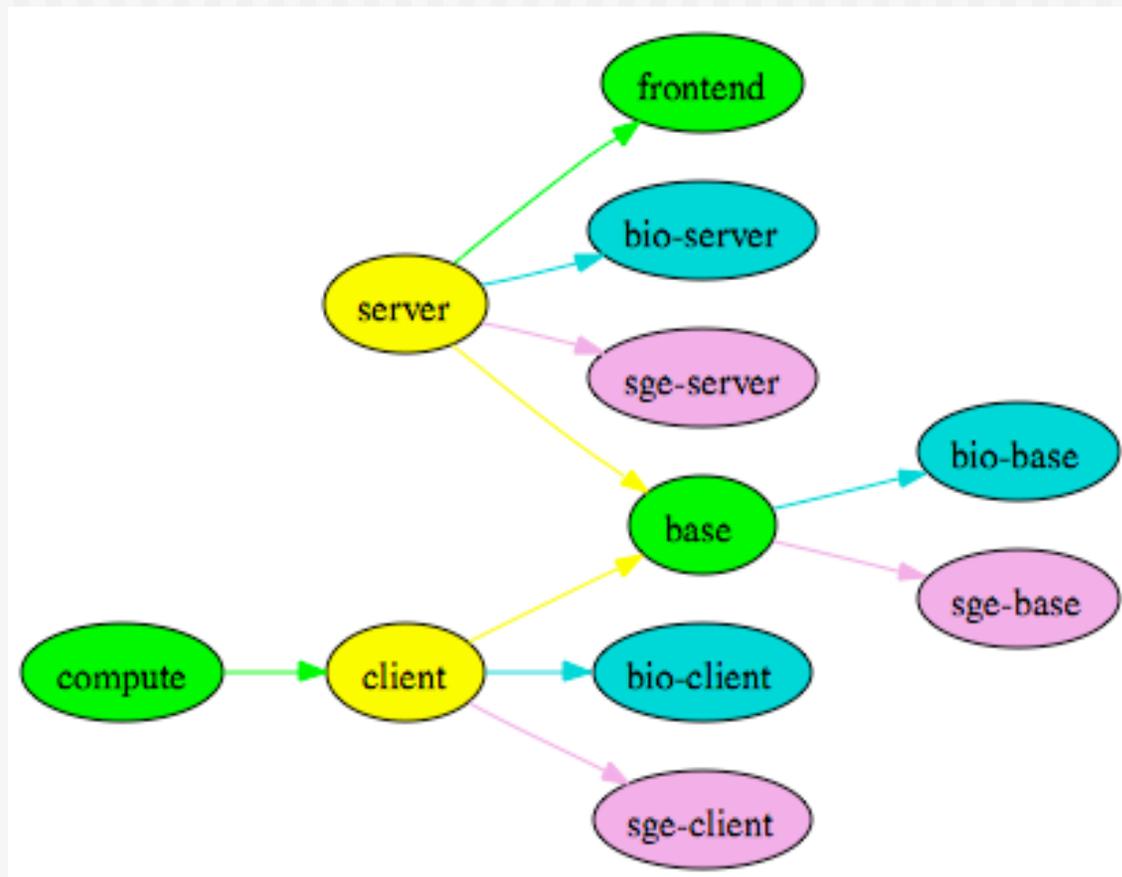
{ base, hpc, kernel, sge }





# Simplified Example

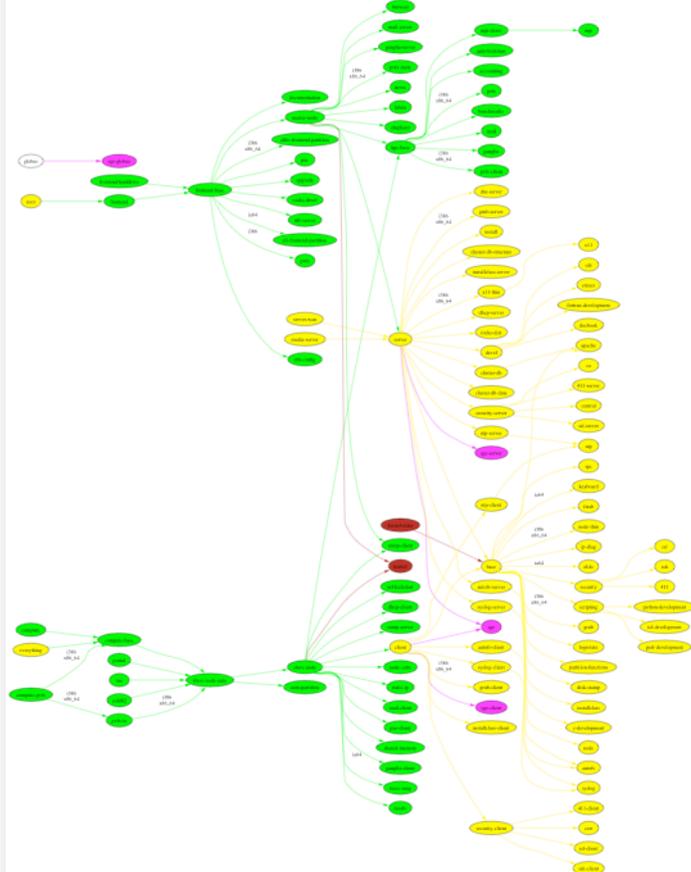
{base, hpc, sge, bio}



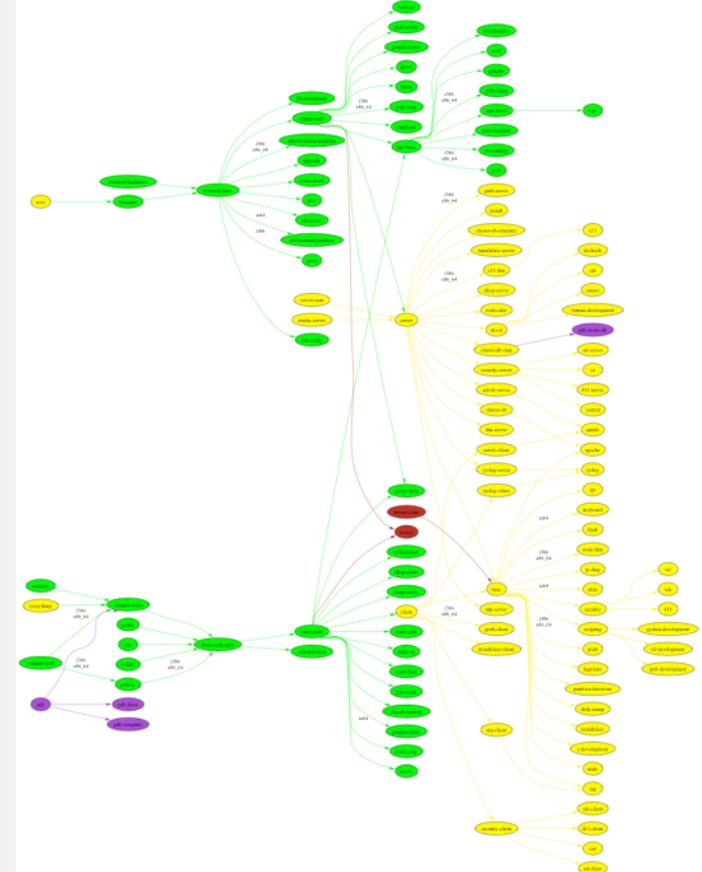


# Two different Clusters

MPI Cluster:::{base, hpc, kernel, sge}



Protein Databank:::{base, hpc, kernel, pdb}





# Where are the Scaling Limits?

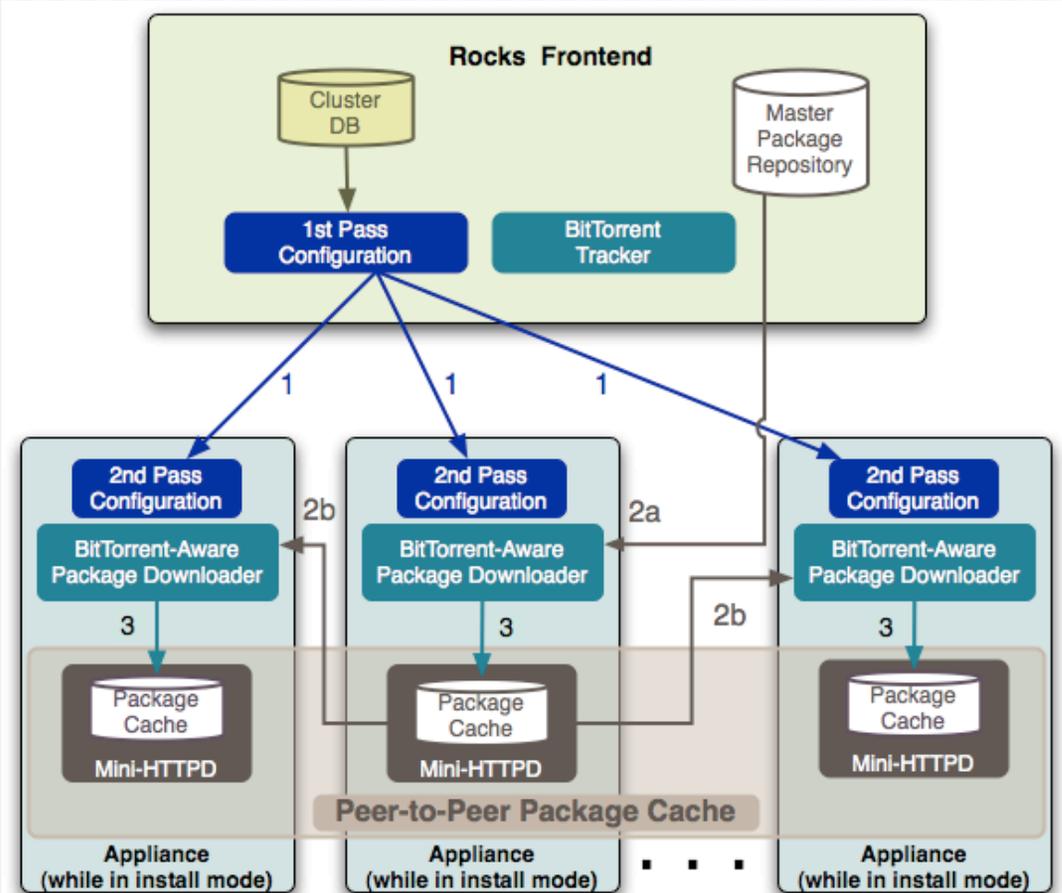
- ◆ Time for Kickstart Generation
  - ⦿ 3 - 4 s / host
  - ⦿  $O(n)$
- ◆ Time to Download Packages
- ◆ Rocks uses HTTP to transport Packages
- ◆ Linux easily serves HTTP files at
  - ⦿ 100MB/sec @ 1Gbit
  - ⦿ 12 MB/Sec@100Mbit
- ◆ Time =  $\langle \#nodes \rangle * \langle \text{total MB packages} \rangle / \text{HTTP Speed}$ 
  - ⦿ Total Packages ~ 350MB

	128 Nodes	1024 Nodes
100 Mbit	3700s (1hr)	9 hours
1 Gbit	460s (8 min)	1 hour



# Avalanche Installer

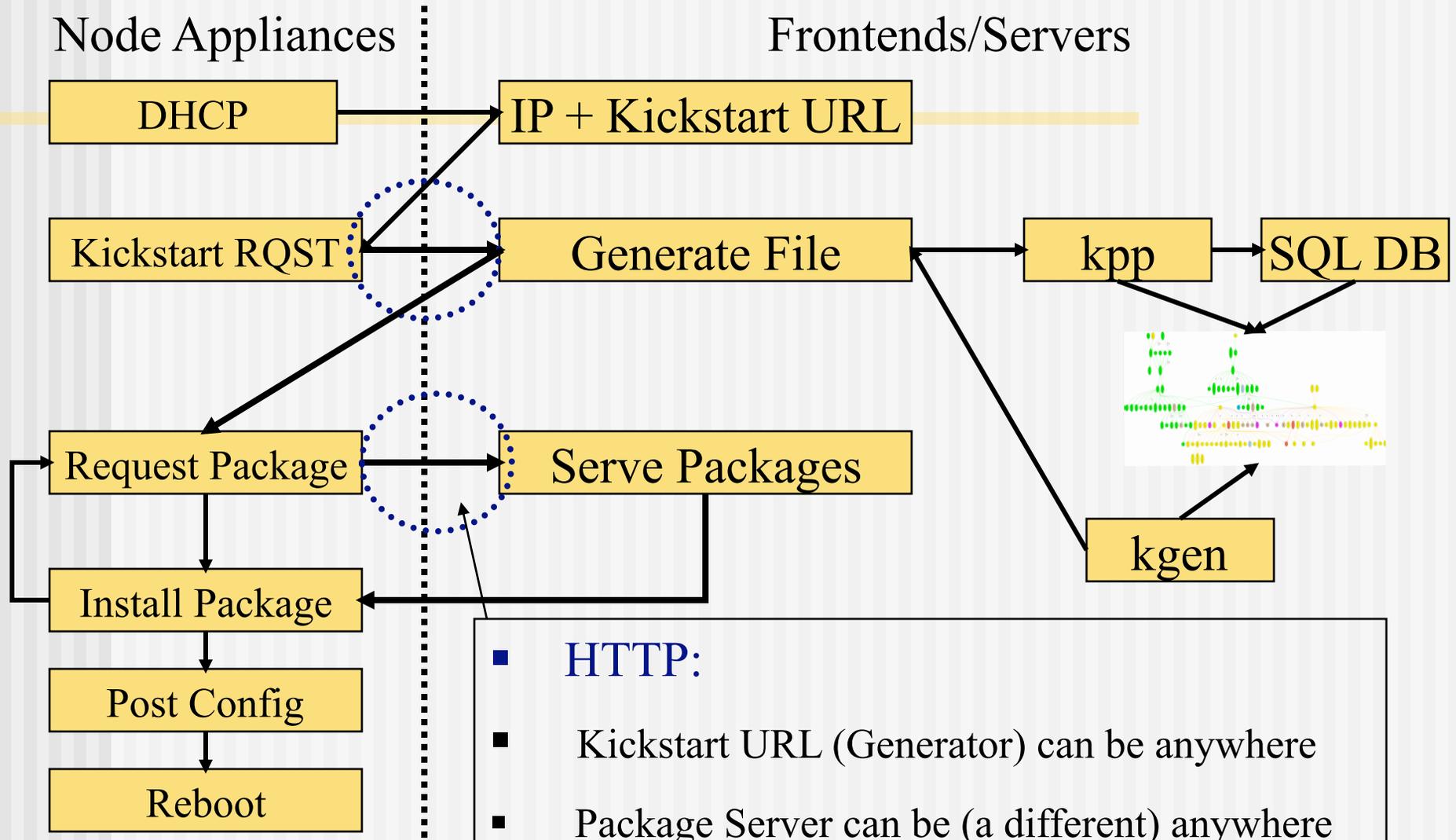
- ◆ Install nodes from a peer-to-peer package cache
- ◆ Takes advantage of switched networks to unload the frontend
- ◆ Kickstart generation is split between frontend and nodes
- ◆ Backoff mechanisms keep the frontend load under control
- ◆ Zero administration





# Pre-Avalanche

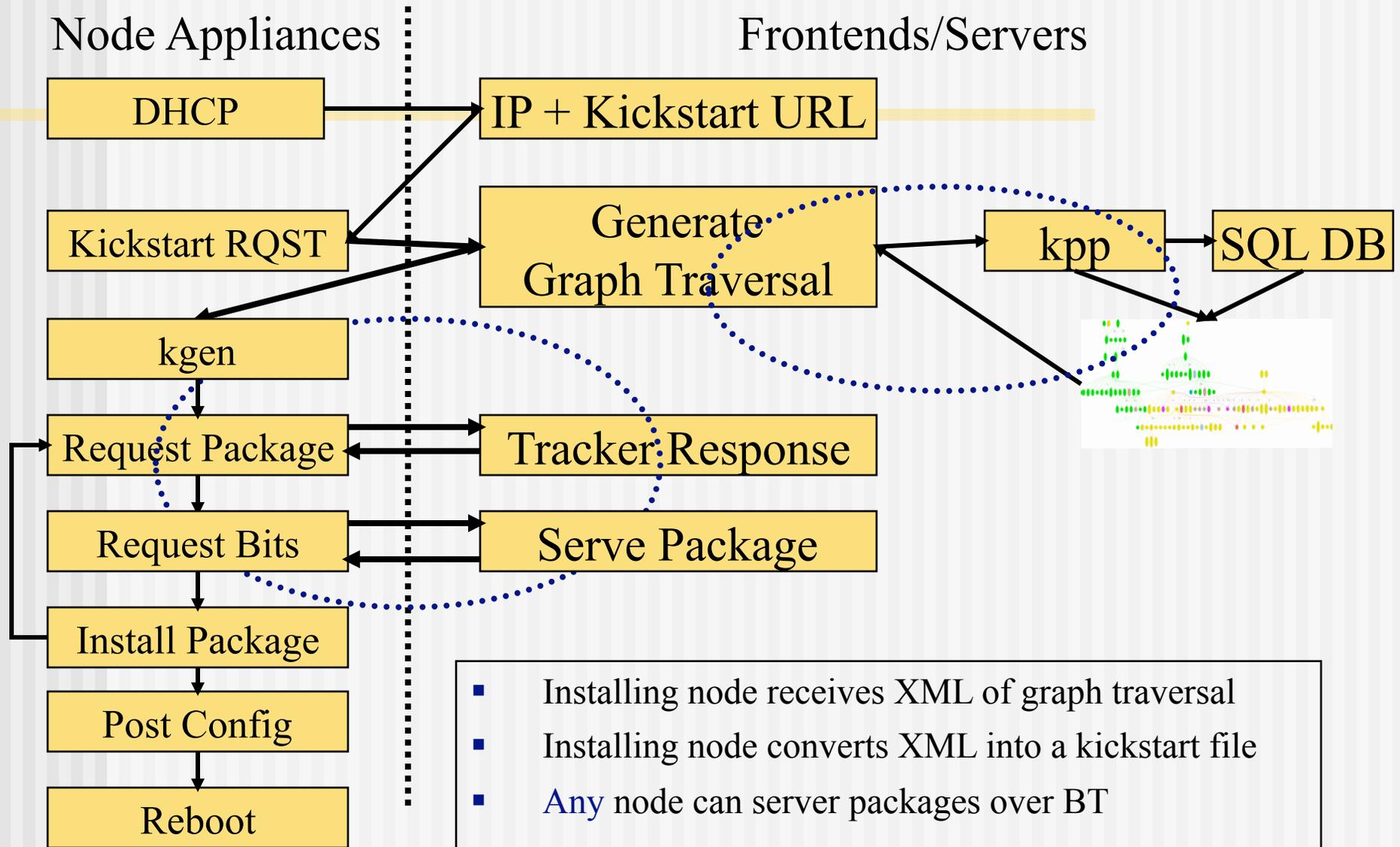
Space-Time and HTTP





# Avalanche

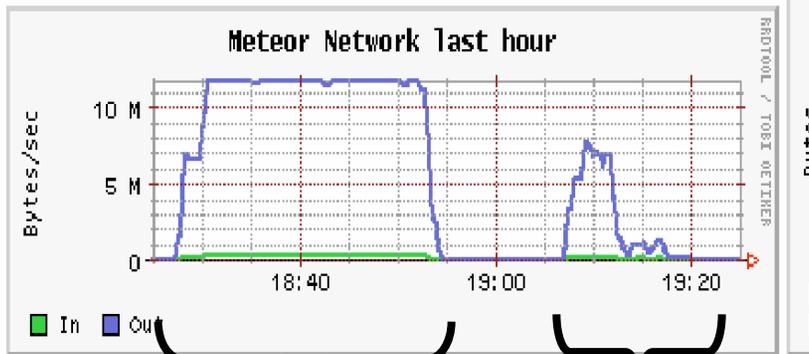
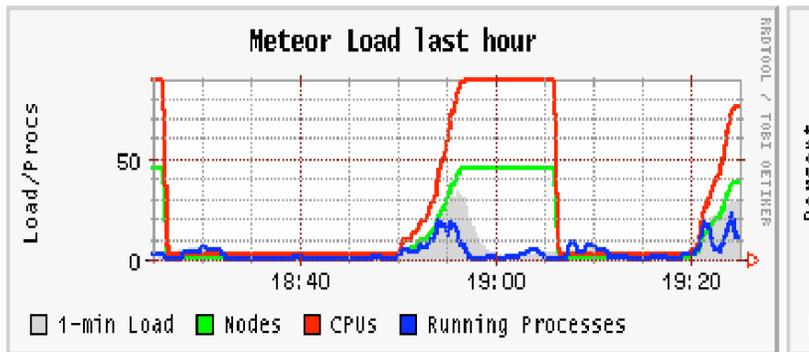
Space-Time and HTTP





# A Glimpse at Performance

## Overview of Meteor



HTTP-  
Only

Avalanche

- ◆ 45 Nodes – 100 Mbit
  - Old and Slow!
  - 350MB (Slim Compute Node)
- ◆ Pre-avalanche:
  - Estimate: 1600s
  - Actual: 1700s
- ◆ Avalanche:
  - Estimate: 900s
  - Actual: 1000s
- ◆ Avalanche is significantly quicker – and reduces load on the frontend



OptIPuter

CalIT/2

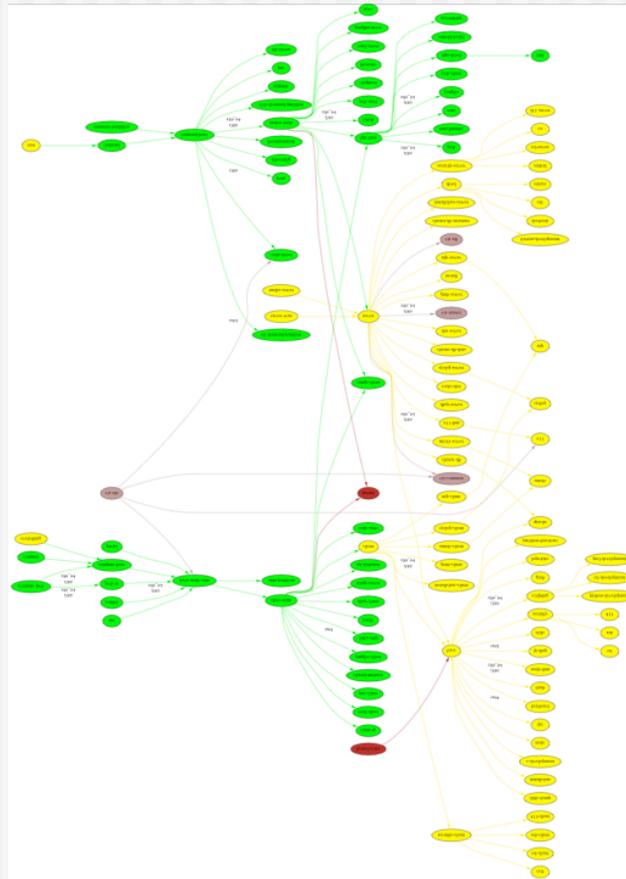
EVL / UIC

# OPTIPORTAL

VIZ ROLL



{ base, hpc, kernel, viz }





# Early Work: NCSA

---

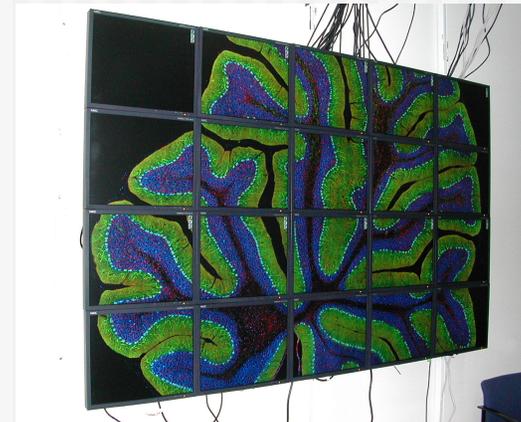
- ◆ LCD Cluster
  - Custom framing
  - One PC / tile
  - Portable (luggable)
  - SC 2001 Demo
- ◆ NCSA Software
  - Pixel Blaster
  - Display Wall In-A-Box
    - OSCAR based
    - Never fully released





# NCMIR

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





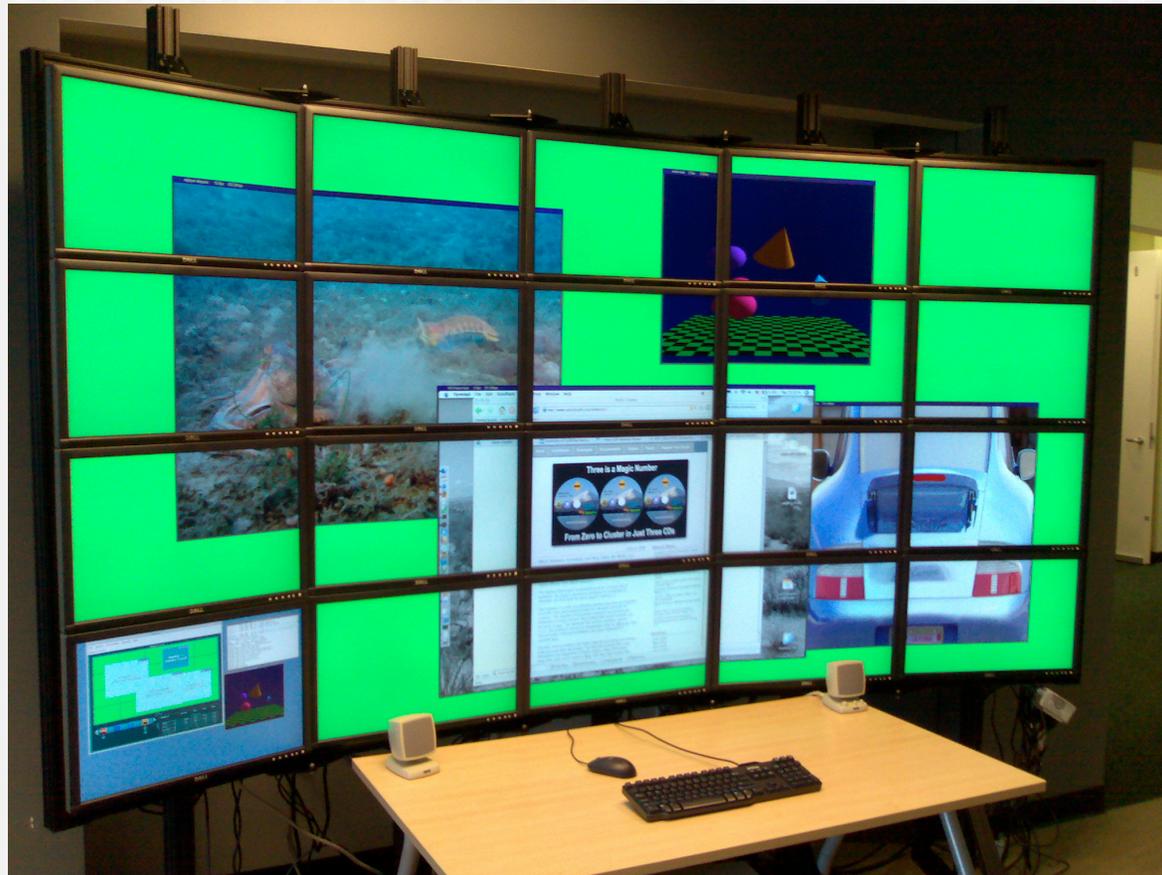
# COTS Vis: GeoWall

- ◆ 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)





# OptIPortal (SAGE)





# One Node per Display





# OptIPortal



5/15/08

© 2008 UC Regents

114



# Nodes Behind the Wall



5/15/08

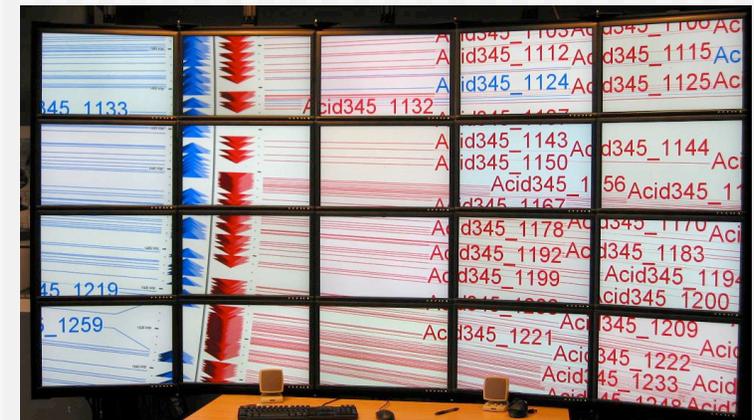
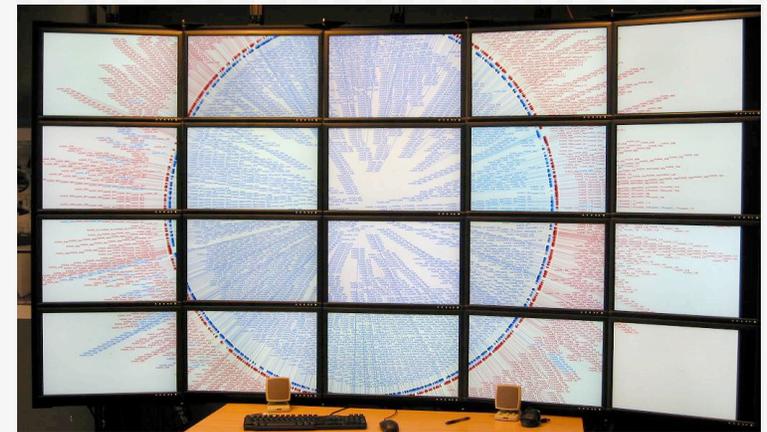
© 2008 UC Regents

115



# Use of OptIPortal 15,000 x 15,000 Pixels to Interactively View Microbial Genome (CGView)

Acidobacteria  
Bacterium Ellin345  
(NCBI)  
Soil Bacterium 5.6 Mb





# Genomic Map (cgview)

---





---

A Meta-Genomic / Bio-Informatic Compute and Data Infrastructure

# CAMERA

# CAMERA: Community Cyberinfrastructure for Advanced Marine Microbial Ecology Research and Analysis

National LambdaRail  
Direct Connect  
Computation and Storage Complex

Funded by: Gordon and Betty Moore Foundation



PI Larry Smarr

Joint Partnership of:

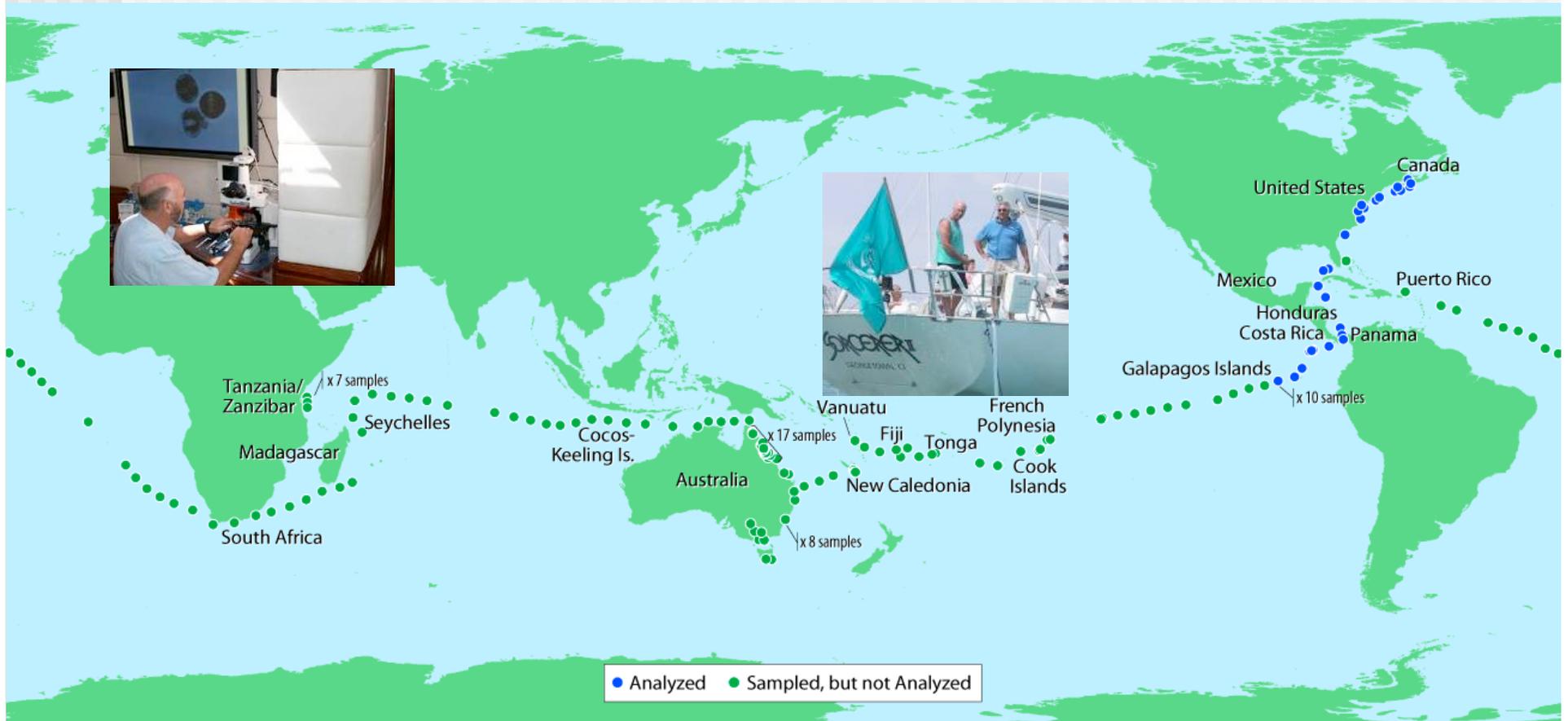


Announced 17 Jan 2006. Public Release 13 March 2007  
\$24.5M Over Seven Years

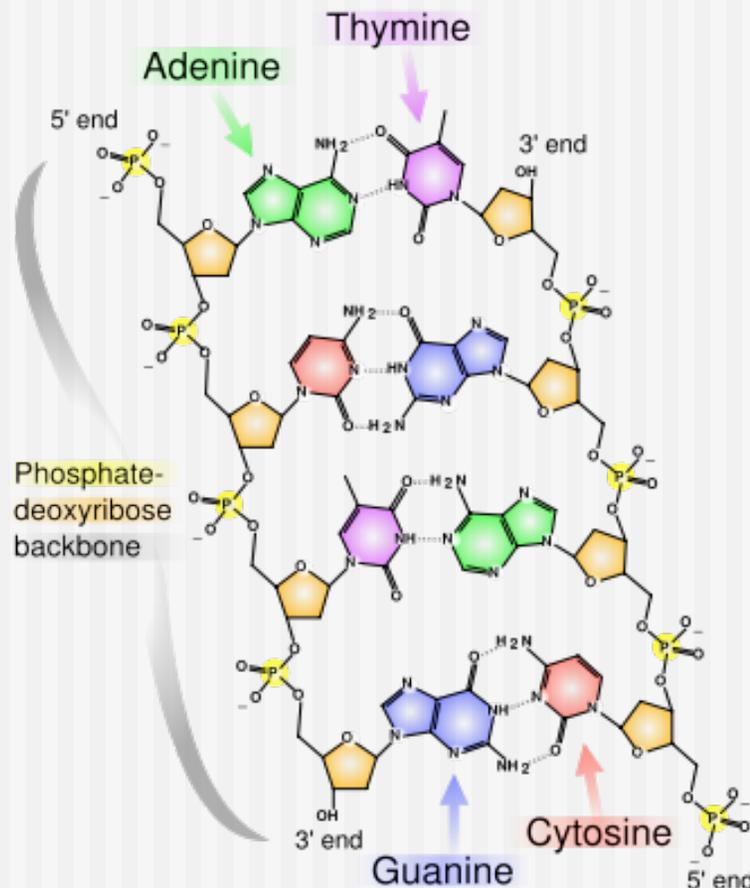
© 2008 UC Regents



# Marine Genome Sequencing Project – Measuring the Genetic Diversity of Ocean Microbes



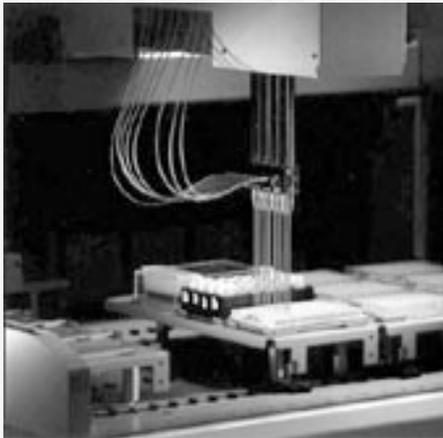
# DNA Basics for Non-Biologists



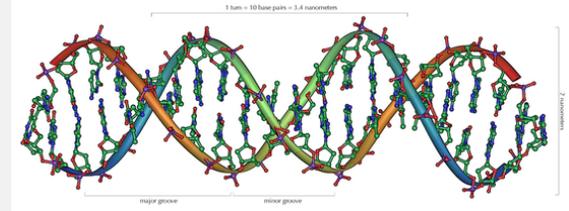
- ◆ Nucleotide bases of DNA
  - ACTG (Adenine, Cytosine, Guanine, Thymine)
  - A Sequence of Bases Forms One Side of a DNA Strand
  - Complementary Bases form the other side of DNA
    - A matches T (pair)
    - C matches G (pair)
- ◆ During cell replication, DNA is “unzipped” . The complementary side can then be replicated perfectly
- ◆ Human DNA is about 3 billion base pairs on 23 Chromosomes



# Sequencers Generate FASTA



```
>JCVI_READ_299 /library_id=JCVI_LIB_GS-00a-01-01-2P5KB-T13532
/template_id=JCVI_TMPL_SHAA001 /sequencing_direction=forward
/sample_id=JCVI_SMPL_1103283000001 /clr_range_begin=85 /clr_range_end=969 /full_length=969
GCGGTTTGG AAGGAACTTCTATT CAGAAAACAAGATAATTTATTTCAATAGGCAAATATATTATCC
AAGAGGTAAAGTCCTTTGTGGCTCTGGCTCAATAAATGCAATGGTCTATGCAAGAGGATTAGAA
ACAGATTATGAGAATTGGGGCACCAATAAGGAATGGAGTTTTGAAAATATAAAAAAATATACAG
ATCTATGGAGCAACAAATAAATGATGATAAAGAATTTCTTACAAAAGAAAAGATTCCAGTAAATAA
TGTAAGTAAGCATCATCATCCAATTTTAGAATATTTTTTTAATGCTAGTAATGAAATTG ....
```



# Bases → Amino Acids

- ◆ Triplets of nucleotide bases are called **codons** and define amino acids.
  - Amino acids are the basic building blocks of proteins
  - There are 20 amino acids, but  $4^3 = 64$  nucleotide combinations.
  - Many amino acids have multiple codons
  - Special codons (called start and stop codons) assist in DNA translation during cell replication.
- ◆ Ambiguity in codon interpretation
  - Depends of where you start
  - For example: GGGAAACC could be:
    - GGG, AAA, CCC (Glycine, Lysine, Proline)
    - CCA, AAC (Glycine, Asparagine)
    - CAA, ACC (Glutamic Acid, Threonine)



# Open Reading Frame → Protein

---

- ◆ Open Reading Frame (ORF)
  - ⇒ portion of an organism's genome which contains a sequence of bases that could potentially encode a protein
    - ATG is a DNA start Codon,
    - An ORF has no stop Codon (TAA, TAG, TGA)
- ◆ When processing a raw DNA read from a Gene Sequencer, you do not know which is the correct reading frame.
  - ⇒ For double-stranded DNA – one of 6 starting positions (3 forward, 3 reverse)



# Sequence Analysis and Comparison

---

- ◆ Raw reads are possible fragments of genes
- ◆ Determining the proper reading frame is difficult
- ◆ Fragment reads are assembled into possible genes (or a complete Genome)
  - ⇒ Shotgun Sequencing, DNA strand is broken up randomly into numerous small segments
  - ⇒ Fragments are then recombined (assembled) to build a whole genome



# Sequence Search and Alignment

- ◆ BLAST (Basic Local Alignment Search Tool)
  - Finds regions of local similarity between sequences.
  - Compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches.
  - Can be used to infer functional and evolutionary relationships between sequences
- ◆ How well do the following sequences align with each other?

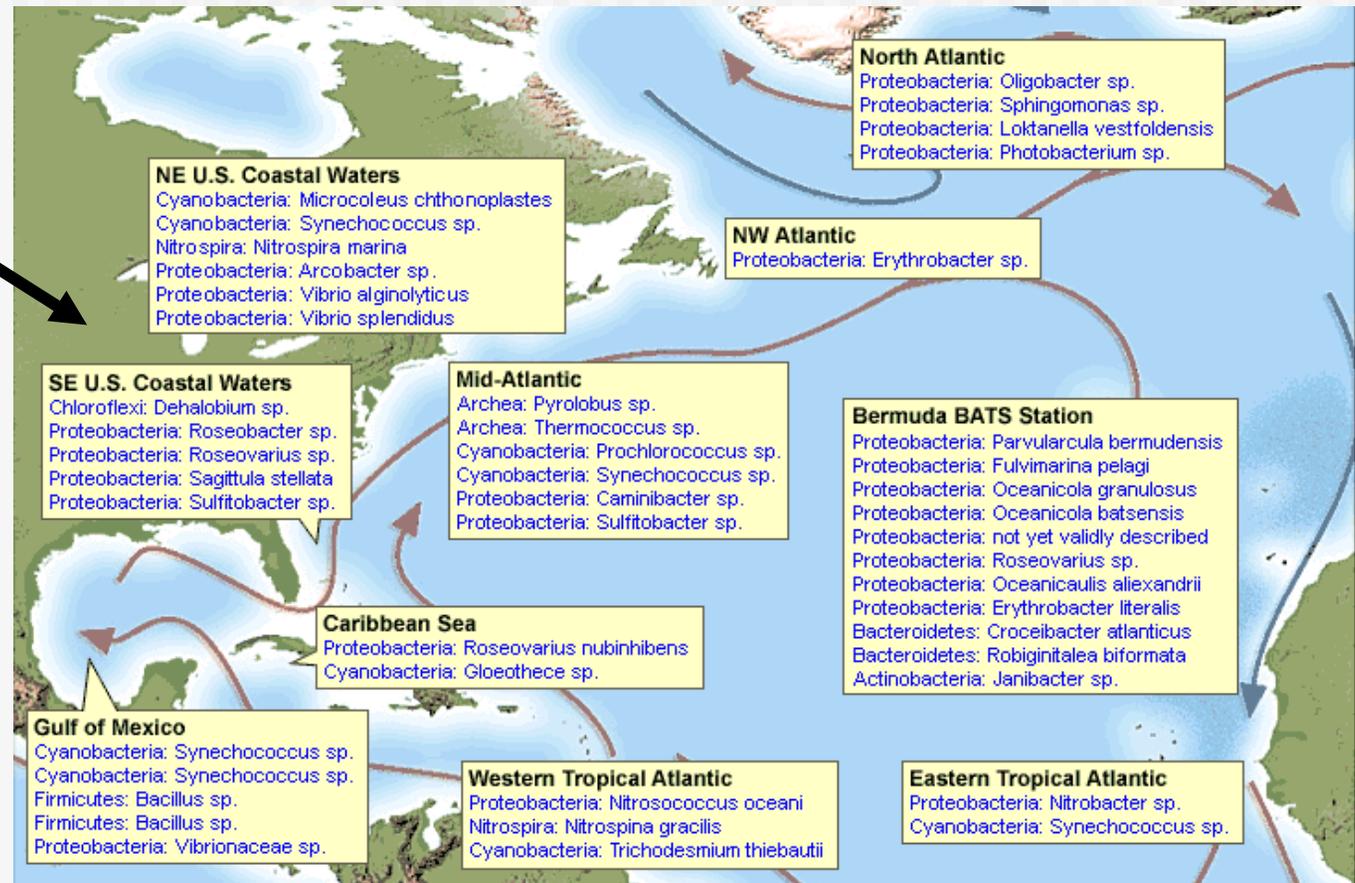
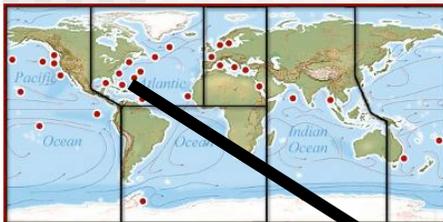
```
TTGGAGTATCTAAACATCCAATCATATTCATTAAAGTCGATTTTCCAGAACCAGAG
GGGCCATAATGAAATATATTCATTTTGATTTATAGTCAAATCGACTCCATCCAAA
GCGCGAACT
```

```
TTGGTGTATCAAGACAACCAATCATATTCATTAATGTTGATTTTCCAGAACCGGAA
GGCCCCATAATAGAAATATATTCATTATGATTAATATTTAAATCAACTCCATCTAAAG
CCCGAACT
```

- ◆ BLAST statistically matches a query sequence against a database of sequences



# Moore Microbial Genome Sequencing Project Selected Microbes Throughout the World's Oceans





# New Application of Shotgun Sequencing

- ◆ Shotgun sequencing is typically performed on single, known organism
  - ⦿ DNA of that organism is cloned for many samples
  - ⦿ It may be difficult/impossible to isolate the DNA of a single microbe
- ◆ Venter Institute applying to whole microbial communities
  - ⦿ A Filtered water sample contains many microbes.
    - This community of microbes is freeze-dried, and sent to Lab
    - This community is then sequenced
  - ⦿ Each individual microbe genome  $O(2M)$  base pairs
  - ⦿ A Read is  $\sim 1000$  base pairs, it is unknown
    - Where in the genome
    - Which microbe is represented
    - Sequencing a Microbial community is believed to be about as complex as building the human Genome
    - It is critical to record the environmental conditions for further comparison



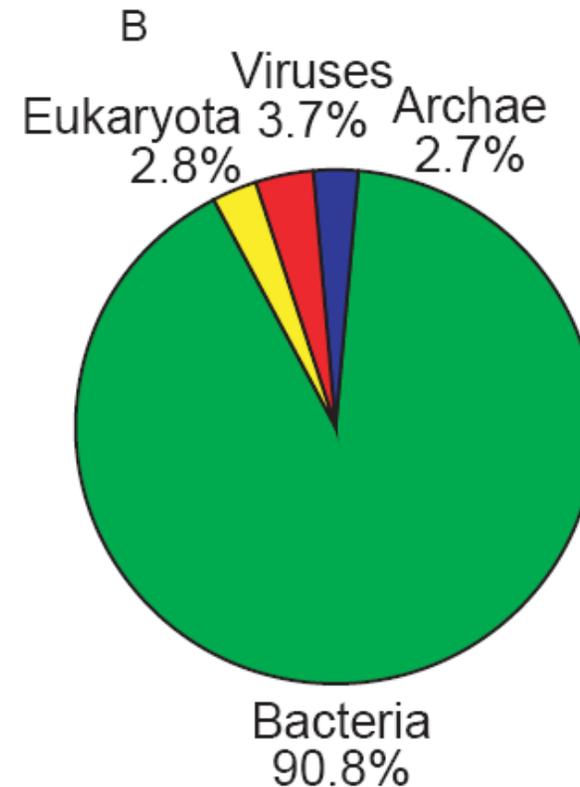
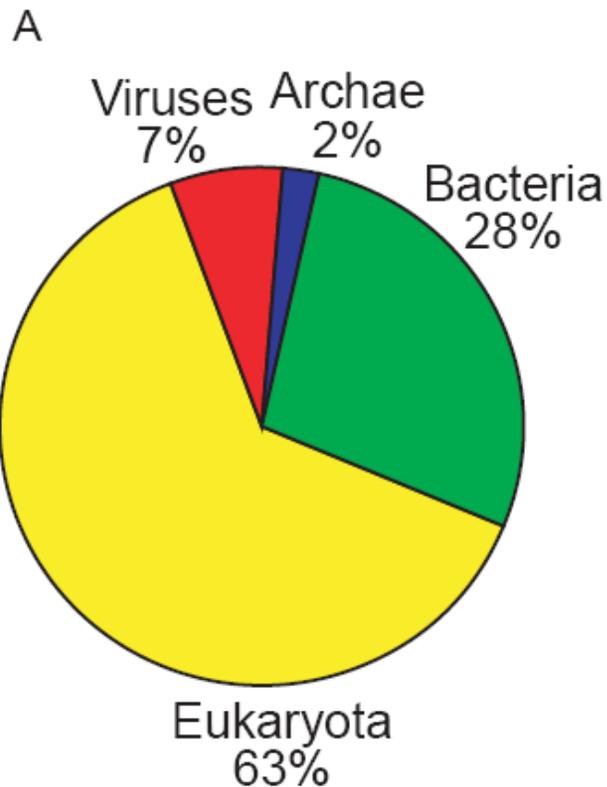
# Some CAMERA Goals

- ◆ Provide an infrastructure where scientists from around the world can perform analysis on genetic communities
  - Global Ocean Sampling (GOS) is the initial large data set
    - ~ 8.5 Billion base pairs of raw Reads
  - **Metadata** is available for samples
    - Saline, Temperature, Geographic Location, Water Depth, Time of Day ...
    - Other metadata will be correlated with samples (e.g. MODIS Satellite)
- ◆ Allow others to search and compare input sequences against CAMERA data.
- ◆ Overall provide a resource dedicated to metagenomics
  - Support new datasets
  - Support new analysis tools and web service



# Global Ocean Survey (GOS) Sequences are Largely Bacterial

~5.6 Million GOS Sequences



~3 Million Previously Known Sequences

5/15/08

© 2008 UC Regents  
Source: Shibu Yooseph, et al. (PLOS Biology in press 2006)



# Meta Data

ID	Sample Loca	Country	Date mm/dd	Time	Location	Sample Dept
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/26/03	03:00	31°32'06"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/26/03	03:35	31°32'06"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/26/03	10:10	31°10'30"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/26/03	10:43	31°10'30"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/25/03	13:00	32°10'29.4"n;	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	2/25/03	17:00	31°32'06"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	5/15/03	11:40	32°10'00"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	5/15/03	11:40	32°10'00"n; 6	5
JCVI_SITE_GS	Sargasso Sea,	Bermuda (UK)	5/15/03	11:40	32°10'00"n; 6	5
JCVI_SITE_GS	Gulf of Maine	USA	8/21/03	06:32	42°30'11"n; 6	1
JCVI_SITE_GS	Browns Bank,	Canada	8/21/03	11:50	42°51'10"n;	1
JCVI_SITE_GS	Outside Halifax	Canada	8/22/03	05:25	44°8'14"n; 6	2
JCVI_SITE_GS	Bedford Basin,	Canada	8/22/03	16:21	44°41'25"n; 6	1
JCVI_SITE_GS	Bay of Fundy,	Canada	8/23/03	10:47	45°6'42"n; 6	1
JCVI_SITE_GS	Northern Gulf	Canada	8/25/03	08:25	43°37'56"n;	1
JCVI_SITE_GS	Newport Harbor	USA	11/16/03	16:45	41°29'9"n; 7	1
JCVI_SITE_GS	Block Island, N	USA	11/17/03	10:30	41°5'28"n; 7	1
JCVI_SITE_GS	Cape May, NJ	USA	11/18/03	04:30	38°56'24"n;	1
JCVI_SITE_GS	Delaware Bay,	USA	11/18/03	11:30	39°25'4"n; 7	1
JCVI_SITE_GS	Chesapeake B	USA	12/18/03	11:32	38°56'49"n; 7	13.2
JCVI_SITE_GS	Off Nags Head	USA	12/19/03	06:28	36°0'14"n; 7	2.1
JCVI_SITE_GS	South of Charl	USA	12/20/03	17:12	32°30'25"n; 7	1
JCVI_SITE_GS	Off Key West,	USA	1/8/04	06:25	24°29'18"n; 8	1.7



512 Processors  
~5 Teraflops  
~ 200 Terabytes Storage

5/15/08



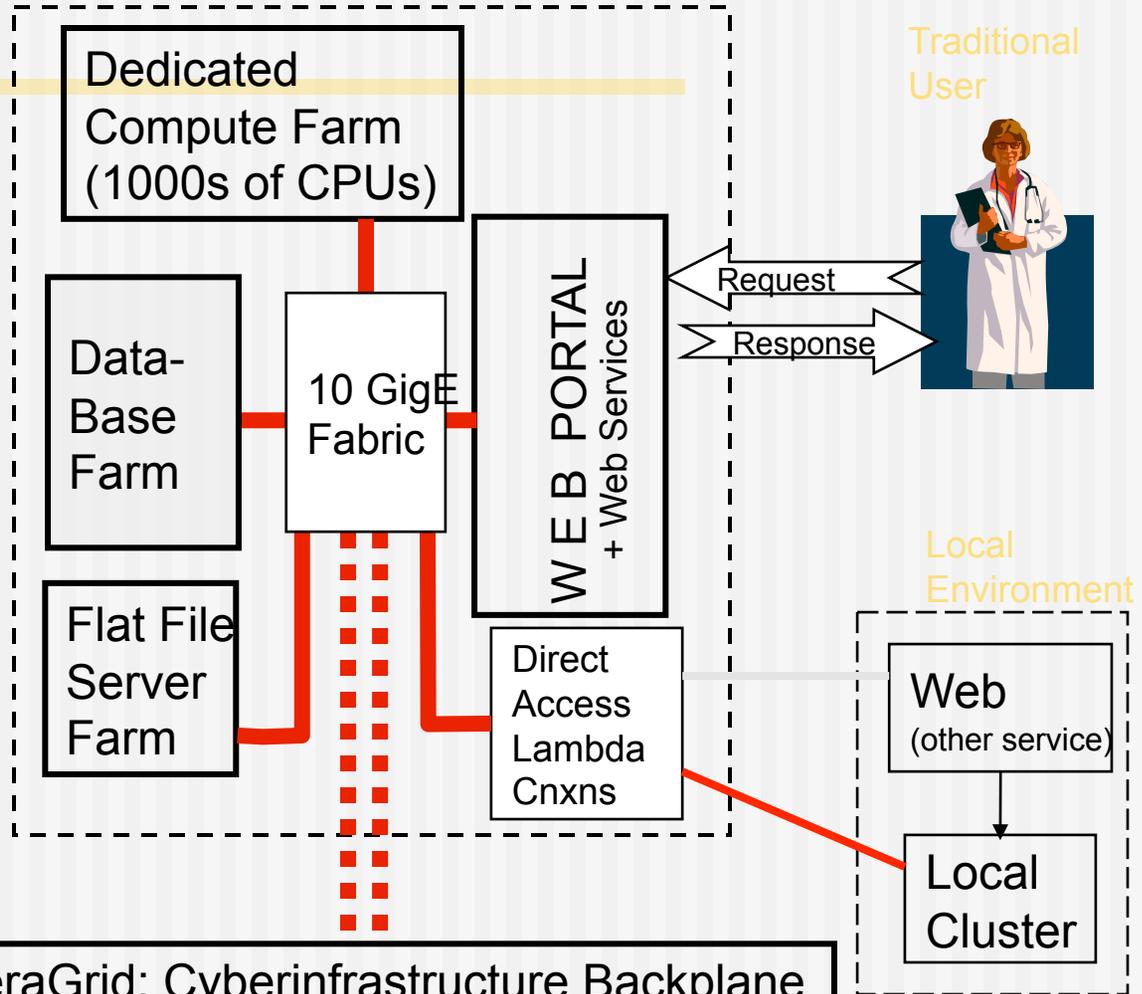
camera J. Craig Venter  
INSTITUTE  
© 2008 UC Regents

132



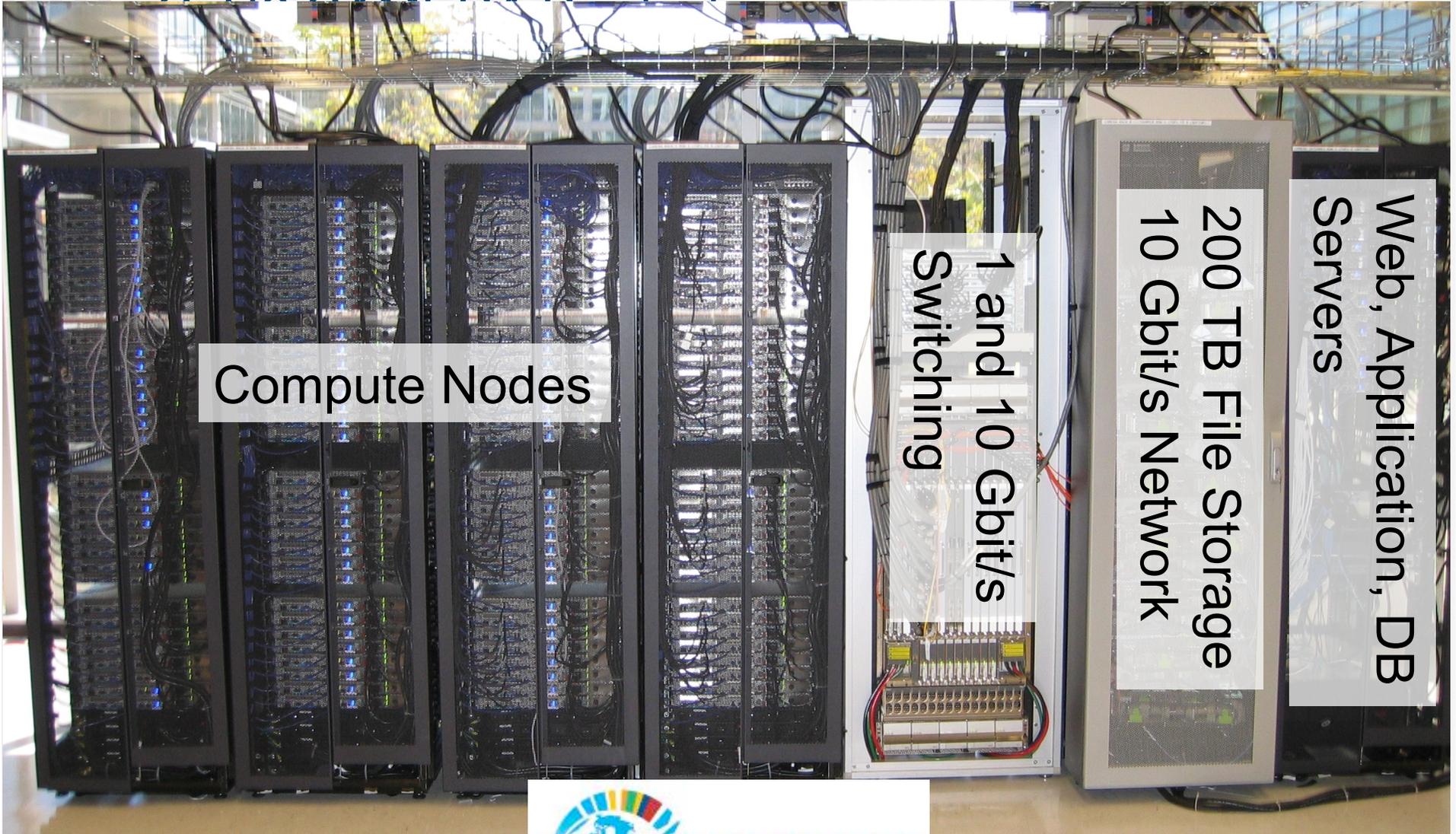
# Calit2's Direct Access Core Architecture CAMERA's Metagenomics Server Complex

- Sargasso Sea Data
- Sorcerer II Expedition (GOS)
- JGI Community Sequencing Project
- Moore Marine Microbial Project
- NASA and NOAA Satellite Data
- Community Microbial Metagenomics Data



**TeraGrid: Cyberinfrastructure Backplane**  
 (scheduled activities, e.g. all by all comparison)  
 (10000s of CPUs)





Compute Nodes

1 and 10 Gbit/s  
Switching

200 TB File Storage  
10 Gbit/s Network

Web, Application, DB  
Servers





# Component Sizing

---

- ◆ ~ 1000 CPU Cores
  - 512 Cores in now, Another 512 – 1024 in about 1 Year
- ◆ ~ 225 TB Raw Disk
  - 200TB now, 400TB in about year
- ◆ 10 Gigabit Ethernet as large subsystem interconnect
  - GigE for all Nodes
  - IB on ½ of Cluster Nodes
  - Force10 E1200 Switch/Router
- ◆ Database servers – as needed



# CAMERA Main Page

www.camera.calit2.net

The screenshot shows the CAMERA website main page. The browser window title is "Camera:" and the address bar shows "http://www.camera.calit2.net/". The website header features the CAMERA logo (a globe with a DNA helix) and the text "camera Marine Microbial Ecology". Navigation links include "HOME", "ABOUT CAMERA", "METAGENOMICS", "RESEARCH", "NEWS", "EVENTS", and "DISCUSSION FORUMS". There are also links for "CONTACT US" and "LOG IN", and a search box with a "go" button.

**WELCOME TO CAMERA!**

CAMERA stands for Community Cyberinfrastructure for Advanced Marine Microbial Ecology Research and Analysis. The aim of this project is to serve the needs of the microbial ecology research community by creating a rich, distinctive data repository and a bioinformatics tools resource that will address many of the unique challenges of metagenomic analysis.

**Version 1.2.6 has been released.**  
We've added 6 new [datasets](#) to our data collection. Some of the key additions are:

- [Mediterranean Gutless Worm Metagenome](#)
- [Acid Mine Drainage Metagenome](#)
- [Waseca County Farm Soil Metagenome](#)
- [and more ...](#)

We've also added some improvements:

- Annotation of all metagenomic datasets with new annotation pipeline
- Improved navigation: more direct access to functions and login no longer required to view projects and publications
- Improved **BLAST** services: support for large jobs, new export formats, and rerun of stored job parameters
- Enhanced [Fragment Recruitment Viewer](#): recruitment of metagenomic reads to over 1,200 reference genomes with addition of new filters and annotation view

Read the [version 1.2.6 release notes](#) for more details of the release.

**What's New**

Have you seen the:

- New [Version 1.2.6](#)
- [Simplified forums](#)
- New [BLAST user guide](#)
- New [Supporting Literature](#)
- New information about [citing CAMERA](#)

**Upcoming Events**

- [Prochlorococcus 20th Anniversary Symposium](#) May 30th, 2008
- [108th ASM General Meeting](#) June 1st, 2008

[More Events](#)

**Metagenomics 2008**  
November 3 - 7, 2008  
Save the date!  
[Metagenomics 2008](#)

**Subscribe** to the CAMERA mailing list to receive notifications of new software releases and data downloads as they become available

**Read this FREE online!**  
[Full Book](#) | [PDF Summary](#)

**Expert Consensus Report**  
Exploring the Microbial Planet

**Funded by the Moore Foundation**  
Gordon and Betty MOORE FOUNDATION



# CAMERA Portal Login

web6.camera.calit2.net

Camera: Login

https://web6.camera.calit2.net/gamasso/gamasso?returnURL=http%3A%2F%2Fweb6.camera.calit2.net%2Fcameraweb%2Fgw%2Forg.jcvi.camera.web.g...

CONTACT US · LOG IN Search go

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

Please enter your username and password to login:

username

password

Not a registered CAMERA user? [Request a CAMERA account](#)

Forgot your password? [Reset it here](#)

Done web6.camera.calit2.net



# BLAST Portal

web.camera.calit2.net

The screenshot shows a web browser window titled "CAMERA Research Home". The address bar contains the URL: <http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.home.Home/Home.htm?#>. The website header features the CAMERA logo (Marine Microbial Ecology) and a navigation menu with links for HOME, ABOUT CAMERA, METAGENOMICS, RESEARCH, NEWS, EVENTS, and DISCUSSION FORUMS. A search bar is located in the top right corner. The main content area is titled "Research Home" and includes a "help" link. Below this, there are five main sections: "BLAST" (with a DNA sequence image), "Projects" (with a diagram of Moore, HOT, and GOS), "Publications" (with Science and PLOS Biol covers), "Samples" (with a map of the Pacific Northwest), and "Fragment Recruitment" (with a colorful bar chart). At the bottom, there is a "Recent BLAST Results" section with a "show" link. The footer contains links for "Help desk" and "Disclaimer", and the version number "Version 1.2.6". The browser status bar at the bottom left shows "Done".



# Projects Include (postgres SQL database)

**Projects**

- Acid Mine Drainage Metagenome
- *Alvinella pompejana* Epibiont Metagenome
- Chesapeake Bay Virioplankton Metagenome (Data Coming Soon)
- Global Ocean Sampling Expedition
- Mediterranean Gutless Worm Metagenome
- Mediterranean Bathypelagic Habitat Metagenome
- Metagenome of Marine NaCl-Saturated Brine
- Microbial Community Genomics at the HOT/ALOHA
- Moore Marine Microbial Sequencing
- Ocean Viruses
- Phosphorus Removing (EBPR) Sludge Community

**Details**

### Global Ocean Sampling Expedition

**Principal Investigator:** J. Craig Venter  
**Funded By:** Moore Foundation, DOE, Venter Science Foundation  
**Organization:** J. Craig Venter Institute  
**Affiliation:**

The broad objective of the global ocean sampling expedition is to expand our understanding of the microbial world by studying the gene complement of marine microbial communities. Marine microbes influence the cycling of carbon (and other elements) in the world's oceans, acting as a biological conduit that transports carbon dioxide from the surface to the deep oceanic realms.

By sequestering carbon from the atmosphere, marine microorganisms (eukaryotes, prokaryotes and viruses) may significantly affect global climate. How they do so, however, is poorly understood, and our attempts to study their activities are limited by our inability to culture the vast majority of them.

One avenue of exploration is to sequence the genomes of marine microbes using a metagenomics approach. In Spring of 2003, the J. Craig Venter Institute conducted a whole environment shotgun sequencing project to study marine microorganisms in the nutrient-poor Sargasso Sea near Bermuda. This study revealed an unforeseen breadth and depth of microbial diversity - about 1,800 different microbial species encoding over 1.2 million genes were discovered, nearly doubling the number of prokaryotic genes available in public databases. Notably, this study expanded our knowledge of ocean photobiology and nutrient pools. Results from the pilot study were reported in *Science* in 2004.

This pilot study served as the springboard for launching a more comprehensive survey of the bacterial, archaeal and viral diversity of the world's oceans. A global circumnavigation aboard the *Sorcerer II*

**More Information**

- >> [Samples](#)
- >> [Publications & Data](#)
- >> [Website](#)
- >> [Contact](#)

click on the image for full collection of articles for Global Ocean Sampling.

The Sorcerer II Research Vessel Courtesy: J. Craig Venter Institute



# Download Data

ftp.camera.calit2.net

The screenshot shows a web browser window displaying the 'camera' website. The URL is <http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.download.DownloadByPubPage/DownloadByPubPage.oa#>. The page title is 'Browse Publications and Data'. The website header includes the 'camera' logo (Marine Microbial Ecology) and navigation links: HOME, ABOUT CAMERA, METAGENOMICS, RESEARCH, NEWS, EVENTS, DISCUSSION FORUMS. A search bar is also present.

The main content area is titled 'Publications and Data'. A dropdown menu shows 'Project: Global Ocean Sampling Expedition'. Below this, there are two tabs: 'Details' and 'Downloads'. The 'Downloads' tab is active, showing a list of publications and their associated data downloads.

**Publications and Data**

Project: Global Ocean Sampling Expedition

**Publications**

- The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pacific  
Douglas B. Rusch, Aaron L. Hal...
- Structural and functional diversity of the microbial kinome  
Natarajan Kannan, Susan S. Tay...
- The Sorcerer II Global Ocean Sampling Expedition: Expanding the Universe of Protein Families  
Shibu Yooseph, Granger Sutton,...
- Environmental Genome Shotgun Sequencing of the Sargasso Sea  
J. Craig Venter, Karin Remingt...
- Assessing diversity and biogeography of aerobic anoxygenic phototrophic bacteria in surface waters of the Atlantic and Pacific Oceans using the Global Ocean Sampling expedition metagenomes  
Natalya Yutin, Marcelino T. Su...
- Gene Identification and Protein Classification in Microbial Metagenomic Sequence Data via Incremental Clustering  
Shibu Yooseph, Weizhong Li, Gr...
- Viral Photosynthetic Reaction Centre Genes and Transcripts in the Marine Environment  
Oded Beja, Yael Mandel-Gutfre...
- The Sorcerer II Global Ocean Sampling Expedition: Metagenomic Characterization of Viruses within Aquatic Microbial Samples

**The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pa**

**Publication Downloads**

[Publication](#)

**Publication Data Downloads**

**Filters** [\[show\]](#)

[\[expand all\]](#) [\[collapse all\]](#)

- [-] Site Metadata
  - [GOS Sites](#) (11 KB)
  - [Chesapeake Bay NSF MOVE](#) (430 B)
- [-] Reads
  - [All GOS Reads](#) (13 GB)
  - [GS000a Shotgun, Sargasso Sea, Station 13 and 11](#) (1.1 GB)
  - [GS000b Shotgun, Sargasso Sea, Station 13 and 11](#) (560 MB)
  - [GS000c Shotgun, Sargasso Sea, Station 3](#) (650 MB)
  - [GS000d Shotgun, Sargasso Sea, Station 13](#) (590 MB)
  - [GS001a Shotgun, Sargasso Sea, Hydrostation S](#) (250 MB)
  - [GS001b Shotgun, Sargasso Sea, Hydrostation S](#) (160 MB)
  - [GS001c Shotgun, Sargasso Sea, Hydrostation S](#) (160 MB)
  - [GS002 Shotgun, Gulf of Maine](#) (150 MB)
  - [GS003 Shotgun, Browns Bank, Gulf of Maine](#) (110 MB)



# User Forums

## web7.camera.calit2.net

Read new messages since my last visit

Forums	Topics	Messages	Last Message
<b>Known Bugs</b> Known bugs, problems, and service downtime notices.	16	23	04/18/2008 18:00:03 <b>abrust</b> →
<b>Known Data Issues</b> Known and reported issues concerning consistency and quality of CAMERA datasets.	1	2	03/20/2008 10:31:19 <b>pgilna</b> →
<b>Report A Problem</b> Report problem with CAMERA system or applications.	29	78	04/07/2008 11:17:58 <b>yoyoman</b> →
<b>Make A Suggestion</b> Provide feedback on new features or suggestions on improving the CAMERA services.	9	14	05/31/2007 05:53:20 <b>CNRS</b> →
<b>Let The Community Know</b> Provide information to the Marine Metagenomics community.	2	3	01/24/2008 15:12:30 <b>genbio</b> →

Who is online
Our users have posted a total of 136 messages We have 374 registered users The newest registered user is <b>crabtree</b>
There are 1 online users: 1 registered, 0 guest(s) [ <b>Administrator</b> ] [ Moderator ] Most users ever online was <b>254</b> on 11/13/2007 03:12:18 Connected users: <b>mjk</b>



# BLAST Query

## searching for ecoli

BLAST Wizard

http://web.camera.calit2.net/cameraweb/gwt/org.jvli.camera.web.gwt.blast.Blast/Blast.htm#BlastWizardUserSequencePage

 **camera** >> research  
Marine Microbial Ecology

CONTACT US LOG OUT SEARCH go

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Sequences > Page 1 of 3

Research Home Tools Projects & Data

### Specify Query Sequences [? help](#)

Specify a nucleotide or peptide query sequences (in [multi-FASTA format](#)):

Back Next

New Sequences Previous Sequences

Sequences Name:  
User sequence 04/23/08 02:15 PM

Enter sequences in multi-fasta format:

```
agcttttcattctgactgcaacgggcaaatatgtctctgtgtgattaaaaaaaaagagtctctgacagcagcttctgaactggt
```

Upload multi-fasta sequences file

Browse... Upload

Done



# Choose Dataset

(~ .5TB FASTA files & SQL DB)

The screenshot shows the BLAST Wizard web interface. The browser address bar displays the URL: <http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.blast.Blast/Blast.htm#BlastWizardSubjectSequencePage>. The page header includes the CAMERA logo (Marine Microbial Ecology) and navigation links: CONTACT US, LOG OUT, and a SEARCH box. A breadcrumb trail reads: >> Research > Tools > BLAST Sequences > Page 2 of 3. The main heading is "Select Reference Datasets to Search Against" with a help link. Below this, there are tabs for "Read Datasets", "Assembled Datasets", "External Datasets", and "All Datasets". The "Read Datasets" tab is active, displaying a table with the following data:

Description	Length	Sequence Type
AcidMine: All Metagenomic Sequence Reads (N)	325,778,024	nucleotide
All Metagenomic 454 Reads (N)	179,275,878	nucleotide
All Metagenomic ncRNAs (N)	33,370,426	nucleotide
All Metagenomic ORF Peptides (P)	4,439,529,911	peptide
All Metagenomic ORFs (N)	13,318,589,733	nucleotide
All Metagenomic Sequence Reads (N)	12,132,838,728	nucleotide
AlvinellaPompejana: All Metagenomic Sequence Reads (N)	290,371,756	nucleotide
DeepMed: All Metagenomic Sequence Reads (N)	7,203,198	nucleotide
EBPRSludge: All metagenomic sequencing reads (N)	221,211,059	nucleotide
FarmSoil: All Metagenomic Sequence Reads (N)	154,475,569	nucleotide
FLAS: All Metagenomic Sequence Reads (N)	2,380,900	nucleotide
GOS: All Metagenomic Sequence Reads (N)	10,634,523,014	nucleotide
GOS: All ncRNAs (N)	33,226,207	nucleotide
GOS: All ORF Peptides (P)	4,395,420,221	peptide
GOS: All ORFs (N)	13,186,260,663	nucleotide
GutlessWorm: All Metagenomic Sequence Reads (N)	314,746,819	nucleotide
HOT: All Metagenomic Sequence Reads (N)	63,837,557	nucleotide
HOT: All ncRNAs (N)	144,219	nucleotide

Below the table, there is a note: "Click once to select. Double click to select and apply." At the bottom of the table area, there are "Back" and "Next" buttons. The footer of the page includes links for "Help desk" and "Disclaimer", and the text "Version 1.2.6". The browser status bar at the bottom shows "Done".



# Name Your Job

BLAST Wizard

http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.blast.Blast/Blast.htm#BlastWizardSubmitJobPage

 **camera** >> research  
Marine Microbial Ecology

CONTACT US LOG OUT SEARCH go

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Sequences > Page 3 of 3

Research Home Tools Projects & Data

**Job Criteria**

Job Name\*:

Query Sequence: User sequence 04/23/08 02:15 PM [\[change\]](#)

Subject Datasets: GOS: All Metagenomic Sequence Reads (N) [\[change\]](#)

**Job Options** [? help](#)

Program:

**Basic Options**

db alignments per query:  [\[A\]](#) [\[V\]](#)

filter low-complexity seq:

evaluate exponent (1Ex):  [\[A\]](#) [\[V\]](#)

lower case filtering:  True  False

**Advanced Options** [\[show\]](#)

[Help desk](#) | [Disclaimer](#) | Version 1.2.6

Done



# Running (Sun Grid Engine Job)

BLAST Wizard

http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.blast.Blast/Blast.htm#SubmitJobWaitPage

**camera** >> research  
Marine Microbial Ecology

CONTACT US LOG OUT SEARCH  go

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Sequences > Running...

Research Home Tools Projects & Data

⚙️ **Your reference sequence is being aligned to your subject sequences.**

**Your job number is 1204962519605577343.**  
You can either wait here for the job to complete, or use the menus above.  
Your job results will be available on the [Job Results](#) page

[Help desk](#) | [Disclaimer](#) | Version 1.2.6

Done



# Waiting...

(Job running on cluster)

Job Results

http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.status.Status/Status.htm#JobResultsPage

**camera** Marine Microbial Ecology >> research

SEARCH CONTACT US LOG OUT go

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Results Research Home Tools Projects & Data

**BLAST Results** [hide] ? help

1 - 5 of 5 [Advanced Sort](#) prev 20 | goto | next 20

	Job Name	Submit Date	Status	Program	# Hits	Subject Sequences	Actions
X	<a href="#">ecoli - reads</a> [edit]	04/23/08 02:17 PM	running	blastn	--	GOS: All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">ecoli</a> [edit]	04/23/08 02:02 PM	completed	blastn	25	GOS: Assembled Sequences (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">bkk 01</a> [edit]	03/21/07 02:25 AM	completed	blastn	0	GOS: All Site-specific Assembled Sequences (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">from bangkok</a> [edit]	03/21/07 01:56 AM	completed	blastn	25	All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">from sofitel</a> [edit]	03/12/07 05:42 PM	completed	blastn	5	GOS: All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>

1 - 5 of 5 Show: [10](#) [20](#) [50](#) < prev 20 | goto | next 20 >

[Help desk](#) | [Disclaimer](#) | Version 1.2.6

Done



# Done

Job Results

http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.status.Status/Status.htm#JobResultsPage



>> research

SEARCH  go

CONTACT US LOG OUT

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Results

Research Home Tools Projects & Data

**BLAST Results** [hide] ? help

1 - 5 of 5 [Advanced Sort](#) prev 20 | goto | next 20

	Job Name	Submit Date	Status	Program	# Hits	Subject Sequences	Actions
X	<a href="#">ecoli - reads</a> [edit]	04/23/08 02:17 PM	completed	blastn	<a href="#">25</a>	GOS: All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">ecoli</a> [edit]	04/23/08 02:02 PM	completed	blastn	<a href="#">25</a>	GOS: Assembled Sequences (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">bkk 01</a> [edit]	03/21/07 02:25 AM	completed	blastn	0	GOS: All Site-specific Assembled Sequences (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">from bangkok</a> [edit]	03/21/07 01:56 AM	completed	blastn	<a href="#">25</a>	All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>
X	<a href="#">from sofitel</a> [edit]	03/12/07 05:42 PM	completed	blastn	<a href="#">5</a>	GOS: All Metagenomic Sequence Reads (N)	<a href="#">Results</a> / <a href="#">Job</a> / <a href="#">Export</a>

1 - 5 of 5 Show: [10](#) [20](#) [50](#) prev 20 | goto | next 20

[Help desk](#) | [Disclaimer](#) | Version 1.2.6

Done



# Results

Job Results

http://web.camera.calit2.net/cameraweb/gwt/org.jcvi.camera.web.gwt.status.Status/Status.htm#

 **research**

CONTACT US LOG OUT

HOME ABOUT CAMERA METAGENOMICS RESEARCH NEWS EVENTS DISCUSSION FORUMS

>> Research > Tools > BLAST Results > BLAST Details

Research Home Tools Projects & Data

**Job Summary** [\[hide\]](#) [back to job results](#) [?](#) [help](#)

Job ID: 1204962519605577343 Submitted: 04/23/08 02:17 PM Query Sequence: [User sequence 04/23/08 02:15 PM](#) Show parameters: [\[view parameters\]](#)  
Job Name: ecoli - reads Program: blastn Subject Sequence: GOS: All Metagenomic Sequence Reads (N)

**Matching Sequences** [\[show\]](#)

**Sequence Alignment** [\[hide\]](#)

Sequence:	<a href="#">JCVI_READ_396311</a>	Score:	121.416	Identities:	70 / 73 (95%)
Sequence Length:	950	Expect:	8.36055e-25	Positives:	0 / 73 (0%)
Alignment Length:	73	Query Begin/End:	128 - 200 (Minus)	Query Gaps:	0
Clear Range:	69 - 778	Subject Begin/End:	375 - 447 (Plus)	Subject Gaps:	0

Query: 200 ATGCGTTTCATGGATGTTGIGTACTCTGTAAATTTTATCTGTCIGTGGCGTATGCCTATATTGGTTAAAGTAT 128  
Sbjct: 375 ATGCGTTTCATGGATGTTGIGTACTCTGTAAATTTTATCTGTCIGTGGCGTATGCCTATATTGGTTAAAGTAT 447

Key: █ match █ similar █ stop codon

**Sequence Geography** [\[hide\]](#)



Map Satellite Hybrid

powered by Google  
Map data ©2008 Tele Atlas, MapLink/Tele Atlas, Europa Technologies, Terms of Usage

>> 14 sample sites are represented in this data set

Done



# Download Datasets

(simple http server)

The screenshot shows a web browser window titled 'CAMERA' with the URL 'http://web4.camera.calit2.net/files/'. The page features a navigation menu with links for HOME, ABOUT CAMERA, METAGENOMICS, RESEARCH, NEWS, EVENTS, and DISCUSSION FORUMS. A search bar is located in the top right corner. The main content area is titled 'SEQUENCE DOWNLOADS' and contains the following information:

CAMERA genomic data sets are listed below. All data is compressed using gzip.

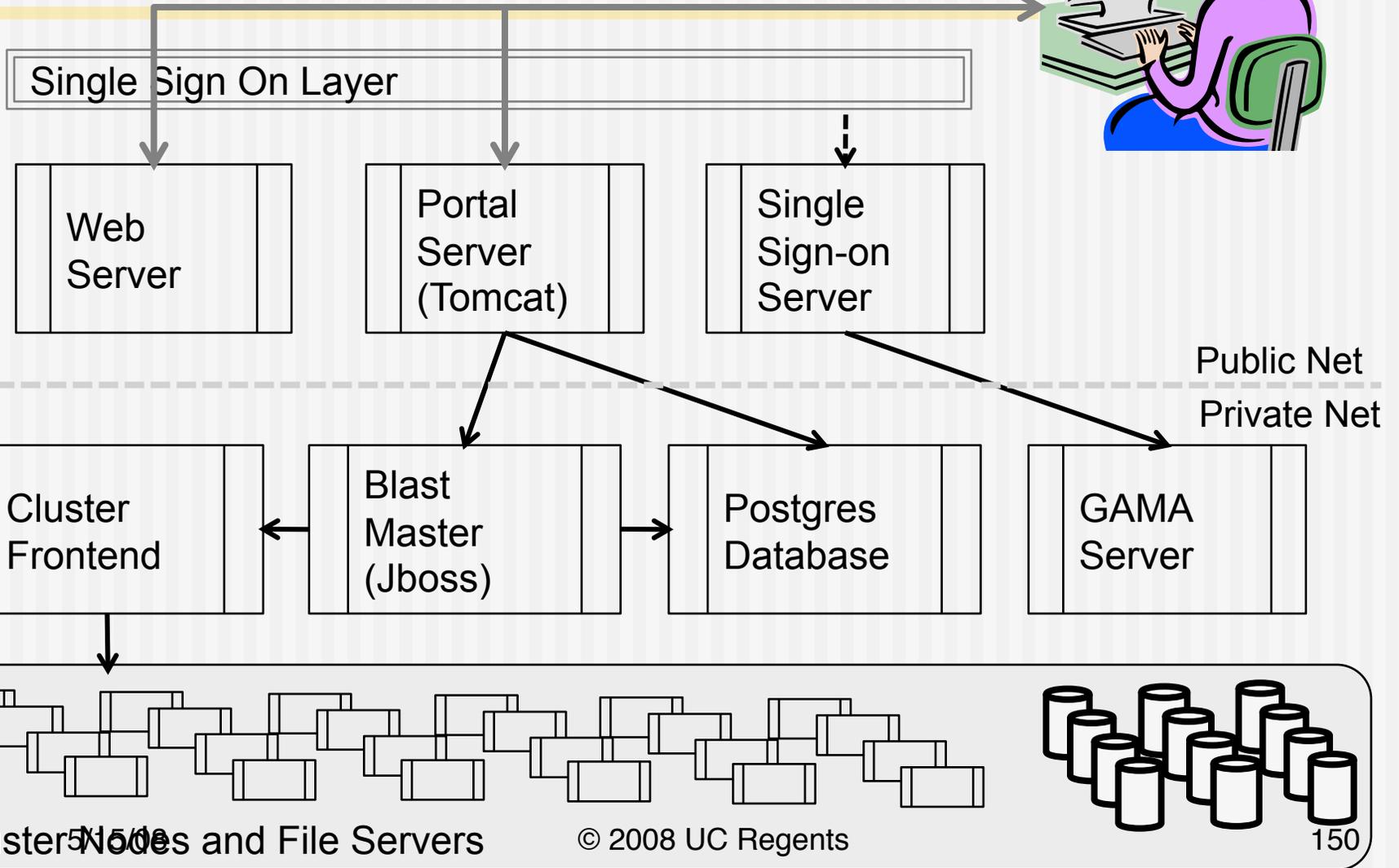
node1015438648499241313.fasta		Available Files	
Name:	GOS: Combined Assembly Proteins (P)	<a href="#">node1015438648499241313.fasta.gz</a>	783.8 MB 2008-03-12 12:58:26
Description:	this file contains proteins predicted via the clustering approach as well as proteins with HMM matches	<a href="#">node1015438648499241313.fasta.gz.md5</a>	67 bytes 2008-03-12 12:58:26
		<a href="#">node1015438648499241313.info</a>	337 bytes 2008-03-12 12:58:26
Sequence Count:	6109770		
Sequence Type:	peptide		
Length:	1,225.15 MB		
Tags:	combined_gos,predicted_protein		
Dataset Node Id:	1015438648499241313		

node1015438648499241314.fasta		Available Files	
Name:	GOS: Combined Assembly Coding Sequences (N)	<a href="#">node1015438648499241314.fasta.gz</a>	1.3 GB 2008-03-12 12:58:44
Description:	this file contains nucleotide coding sequences corresponding to the proteins predicted via the clustering approach as well as proteins with HMM matches	<a href="#">node1015438648499241314.fasta.gz.md5</a>	67 bytes 2008-03-12 12:58:44
		<a href="#">node1015438648499241314.info</a>	394 bytes 2008-03-12 12:58:44
Sequence Count:	6109770		
Sequence Type:	nucleotide		
Length:	3,666.29 MB		
Tags:	combined_gos,predicted_gene		
Dataset Node Id:	1015438648499241314		



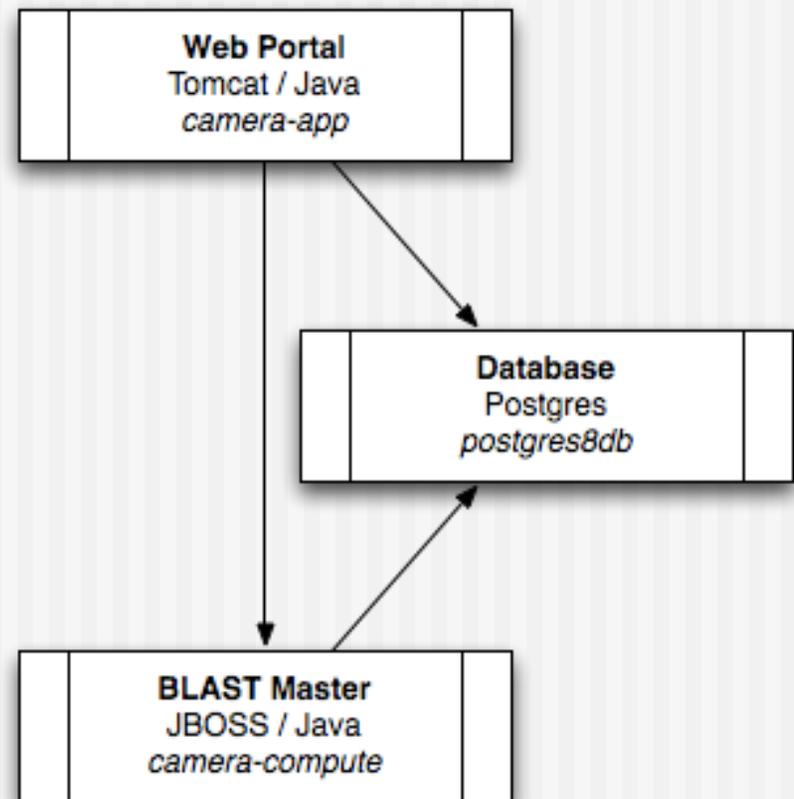
# Logical Layout of Servers





# Portal/Database/Blast Server Interactions

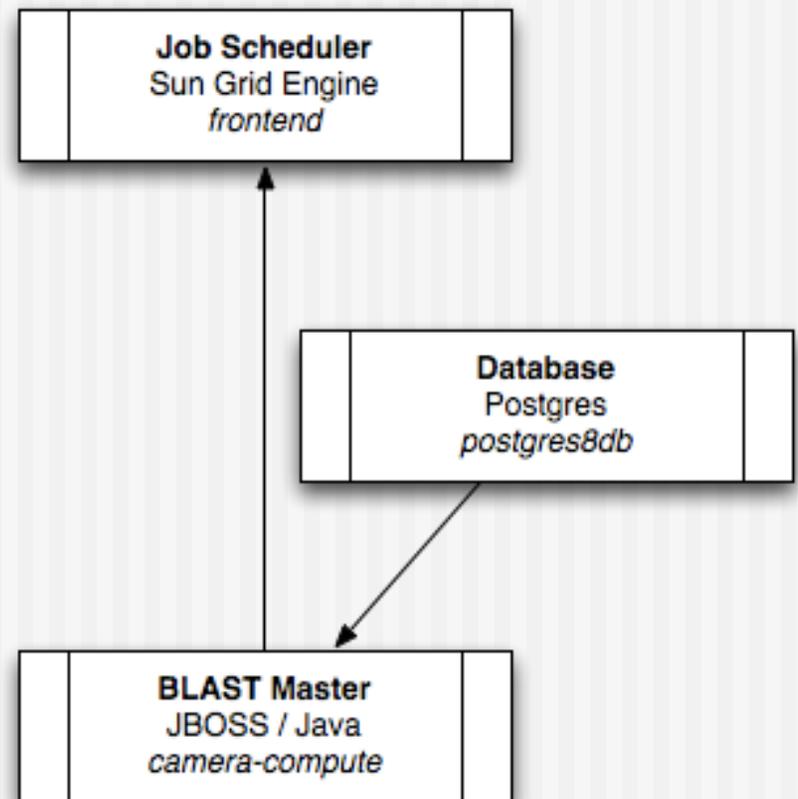
- ◆ Portal is a Tomcat Java application. (camera-app appliance)
  - All data presented in the portal is kept in a Postgres Database
- ◆ When a BLAST job is submitted, control of the job is handed to a JBOSS application (Camera-Compute)
- ◆ The Camera-Compute formats a batch job and submits to the cluster frontend via DRMAA interface





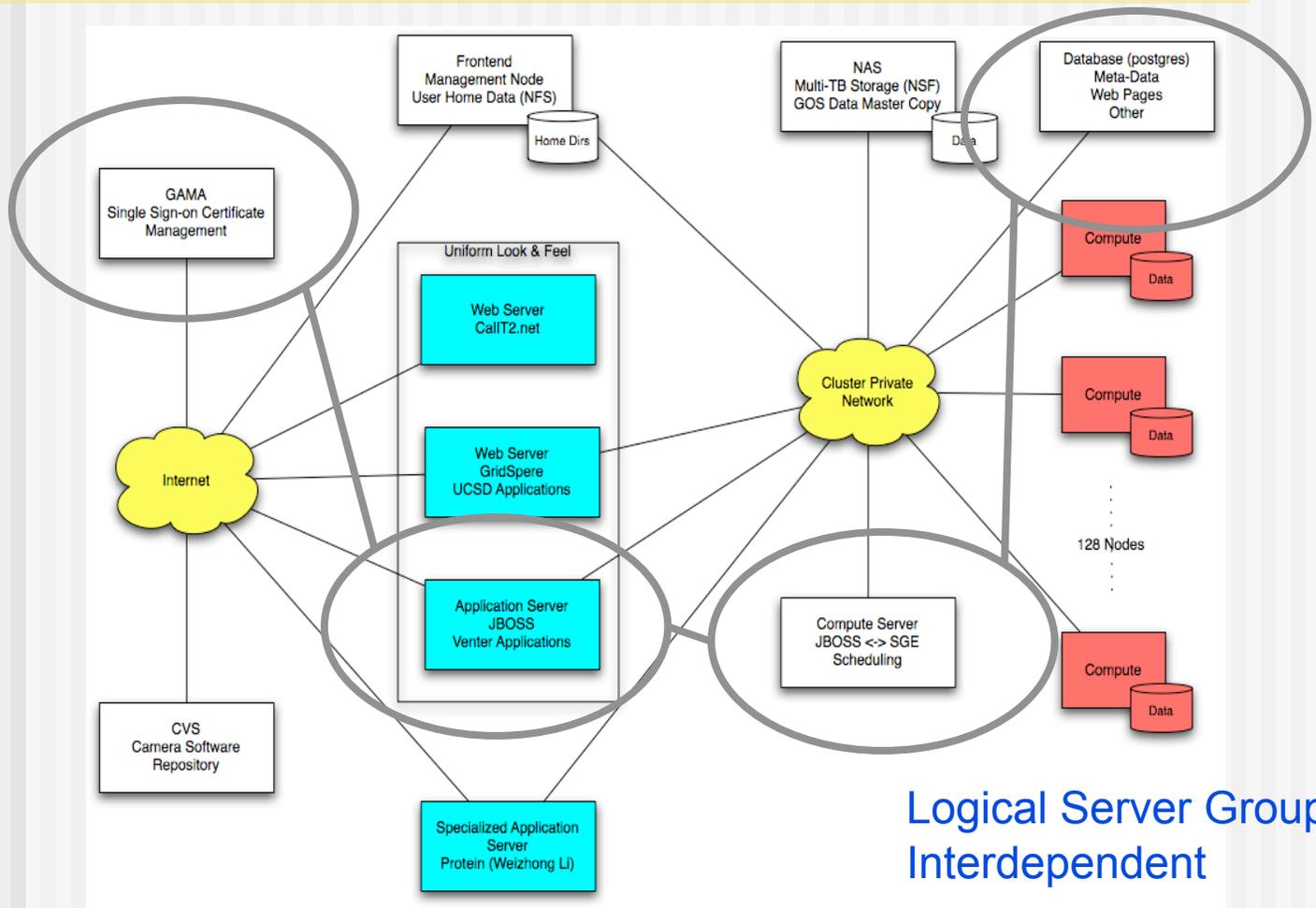
# Compute Server Segmenting

- ◆ Blast Master receives job request from Portal Server
- ◆ Reads from Postres Database how Blastable files have been segmented
  - All ORFs – 134 Segments
  - All Reads – 84 Segments
- ◆ An SGE Array Job is send to the queue master for submission to compute cluster
- ◆ Raw Results are stored in the file system





# CAMERA System Architecture. A Modestly Complex Grid Endpoint



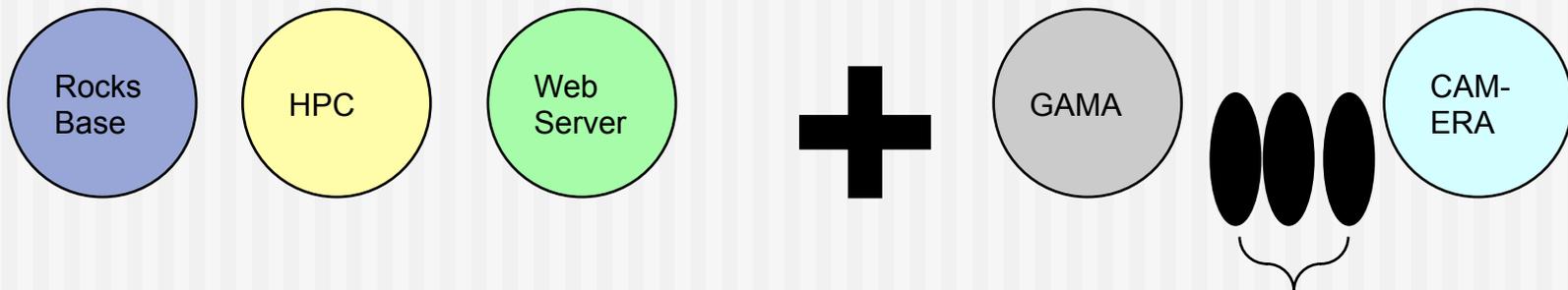
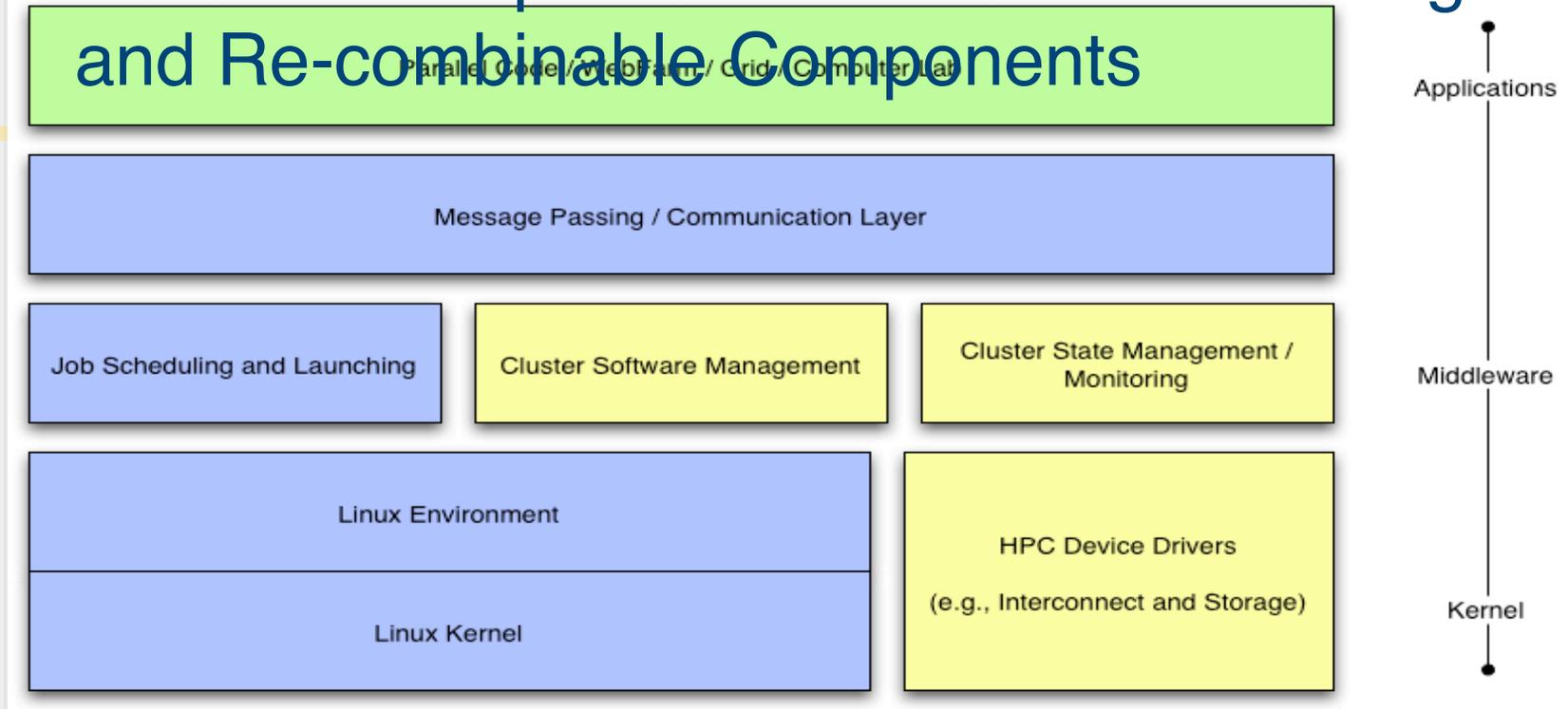


# Using Rocks to Manage Complete Deployment

- ◆ Production CAMERA is deployable by **one system engineer**
- ◆ We have to support three phases of software development
  - Development
  - Staging (Testing)
  - Production
- ◆ Development is done on a completely separate cluster
  - Development in rockscluster.org domain
  - Production/Staging in camera.calit2.net domain
  - All share the same GAMA (Certificate Authority) server

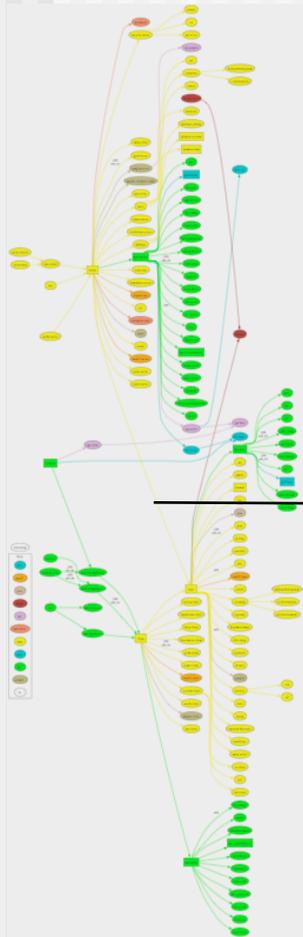


# Rolls Break Apart Software Stacks into Logical and Re-combinable Components



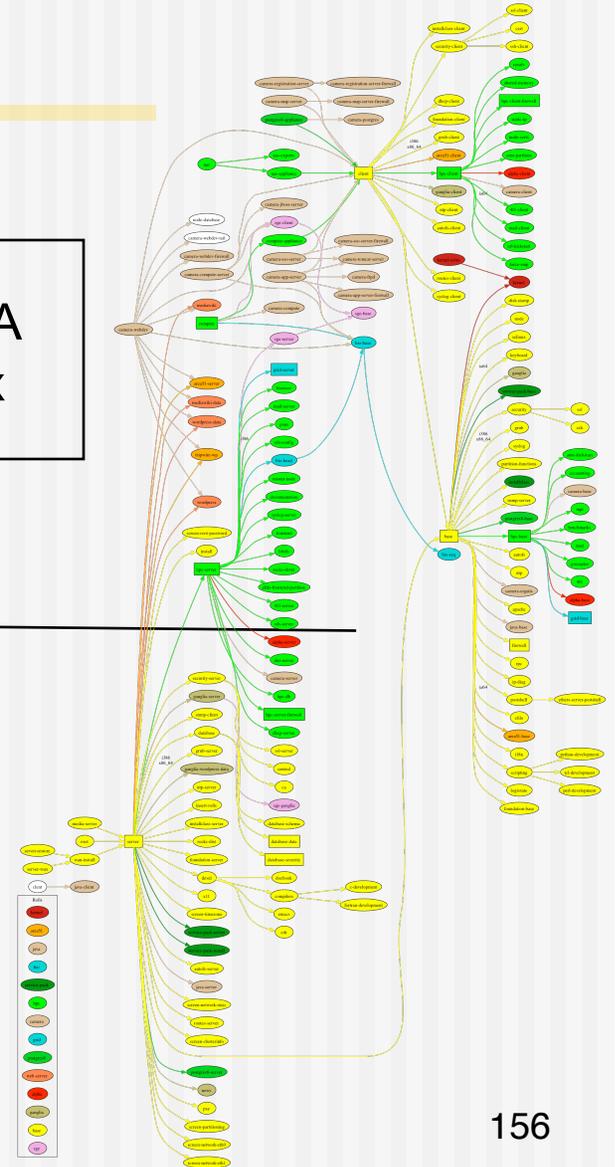


# Why CAMERA is not too different



Standard  
Compute  
Cluster

CAMERA  
Complex

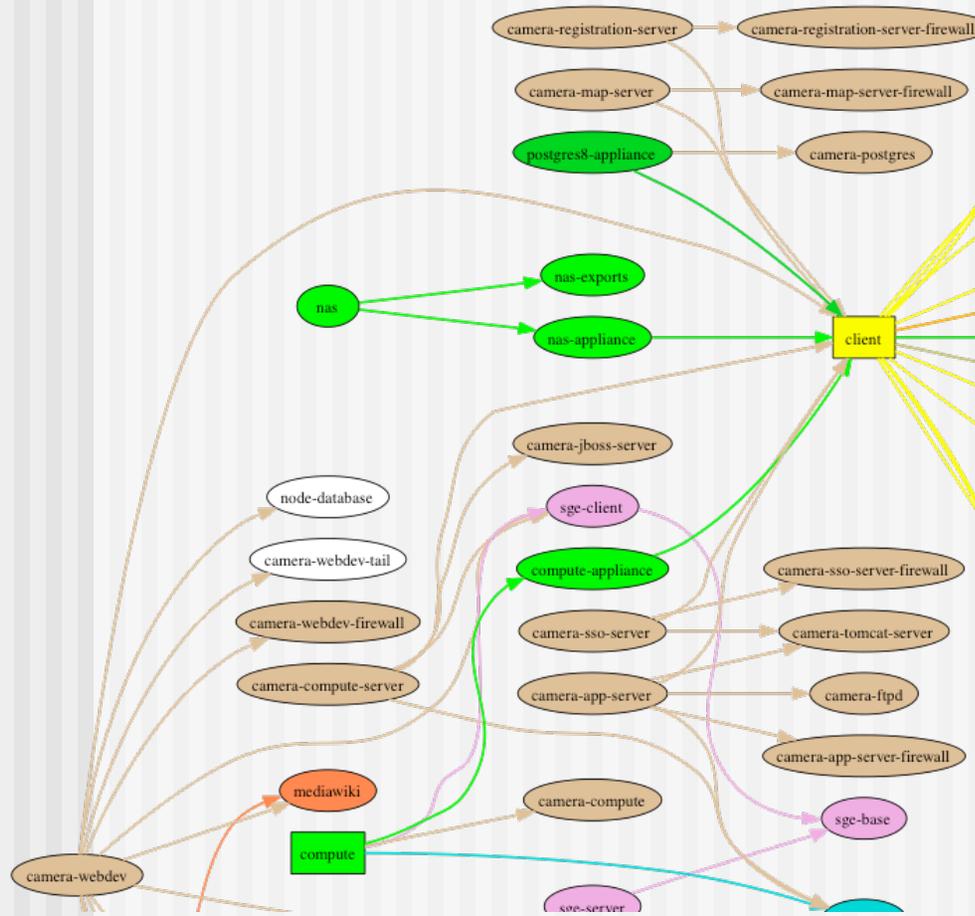


Configuration  
Graphs  
Describe how to  
build a complex  
of clusters





# How the Different Appliances Are Defined



- ◆ Appserver
  - ⊃ FTP server
  - ⊃ App server Firewall
  - ⊃ Tomcat Server
  - ⊃ Bio Roll Base
  - ⊃ Client
- ◆ Camera Compute Server
  - ⊃ SGE Client
  - ⊃ Jboss Server
  - ⊃ Bio Roll Base
  - ⊃ Client
- ◆ Cluster Compute Node
  - ⊃ Bio Roll Base
  - ⊃ Camera Compute addons
  - ⊃ SGE Client
  - ⊃ Compute-Appliance → Client



# Inside the Camera-app-server - Packages

```
<kickstart>
```

```
<package>cameraweb</package>  
<package>cameraweb-jaas</package>  
<package>gama-sso-client</package>  
<package>postgres8</package>  
<package>libungif</package>  
<package>fonts-xorg-75dpi</package>  
<package>Xaw3d</package>  
<package>emacs-common</package>  
<package>emacs</package>  
<package>emacs-el</package>  
<package>emacs-nox</package>
```

```
</kickstart>
```

- ◆ 3 Camera Specific
- ◆ One generic, but different version
- ◆ Remainder are standard RedHat packages



# Inside the Camera-app-server - Config

```
<post>
<file name="/etc/fstab" mode="append">
<var name="GAMASSO_filestore-server"/>:/var/tmp/gama    /var/tmp/gama    nfs
    ro,soft,actimeo=3          1 1
</file>
/sbin/chkconfig --add gama-sso-client

<file name="/opt/tomcat/webapps/cameraweb/WEB-INF/classes/camera.properties">
<eval>
file=/opt/tomcat/webapps/cameraweb/WEB-INF/classes/camera.properties
/opt/rocks/bin/rocks list camera javaprops $file
</eval>
</file>
chown 412.412 /opt/tomcat/webapps/cameraweb/WEB-INF/classes/camera.properties
```

- Mount the single-sign-on file system from the SSO server
- Localize the Cameraweb java application



# How to Handle Java Application Location

---

- ◆ Many Java applications have properties files that are read to define localization
  - Where to find files
  - Location of support servers
  - Debug level
- ◆ The CAMERA Blast Portal has over 50 localized variables
  - Editing manually can lead to higher error rates
- ◆ Variables are different for Production, Development, and Staging



# Camera.properties File Rewriting

## Original file with paths defined in developers environment

```
...  
# DRMAA Blast Merge/Sort settings  
BlastServer.GridJavaPath=/usr/local/bin/java  
BlastServer.GridMergeSortClassPath=/project/camera/runtime-shared/1.0.x/jars/  
camera-blast-grid-merge.jar  
BlastServer.GridMergeSortProcessor=org.jcvi.camera.shared.blastxmlparser.Blast  
GridMergeSort  
BlastServer.GridMergeSortMinimumMemoryMB=384  
...
```

## Deployed file with paths defined for production environment

```
...  
# DRMAA Blast Merge/Sort settings  
BlastServer.GridJavaPath=/usr/java/jdk1.5/bin/java  
BlastServer.GridMergeSortClassPath=/opt/camera/camera-blast-grid-merge.jar  
BlastServer.GridMergeSortProcessor=org.jcvi.camera.shared.blastxmlparser.Blas  
tGridMergeSort  
BlastServer.GridMergeSortMinimumMemoryMB=384
```



# Generic Properties (Not Just Java)

- ◆ The Rocks Database is the Master Record for variables
  - Any variables set in a configuration file are defined here
- ◆ Key, Value table (app\_globals) has four critical fields
  - Appliance, Service, Key, Value
- ◆ For example
  - 0 (0 = valid for any appliance, >0 for a particular appliance only)
  - /opt/tomcat/webapps/cameraweb/WEB-INF/classes/camera.properties
  - BlastServer.GridJavaPath
  - <indirect service="CAMERAETC" component="JavaPath"/>



# <indirect> Construction

- ◆ In Building CAMERA we found that
  - Many different (non-Rocks) components needed the identical value
  - E.g.
    - Cameraweb application needed name of SSO server
      - /etc/fstab also needed name of SSO server
    - Cameraweb needed name of Database server and connection info
      - So does the Jboss Compute server
      - So does an Administrative (read-only) console
  - Production/Staging environments required small changes
    - Staging Server needed it a Staging (test) database
    - Most other variables were identical with production version
- ◆ <indirect>
  - In the “value” part of the database.
  - Acts a data pointer to another [Service, Key, Value] triplet
  - When a non-zero appliance is supplied, can overwrite for the generic version



# Using <indirects>

- ◆ Rocks de-references all indirects at kickstart generation – application/roll developer need not know about indirects.
- ◆ Available as `<var name="Service_Component">`
  - Appliance-specific overrides are handled automatically
- ◆ From appserver definition

```
<var name="GAMASSO_filestore-server"/>:/var/tmp/gama /var/tmp/gama nfs ro,soft,actimeo=3
  1 1
</file>
```

service	Component	value
GAMASSO	filestore-server	<indirect service="GAMASSO" component="PrivateHostname"/>
GAMASSO	Debug	false
GAMASSO	FileDirectory	/var/tmp/gama
GAMASSO	CookieName	gamasso
GAMASSO	PrivateHostname	<indirect service="CAMERAHOSTS" component="SSOserverPrivate"/>
GAMASSO	SiteDomainname	camera.calit2.net

- ◆ Filestore\_server is double indirect as
  - CAMERAHOSTS, SSOserverPrivate



# Production and Staging Hosts

Membership	service	Component	value
0	CAMERAHOSTS	AppServer	web.camera.calit2.net
13	CAMERAHOSTS	AppServer	web3.camera.calit2.net
14	CAMERAHOSTS	AppServer	web3.camera.calit2.net
0	CAMERAHOSTS	ComputeServerPrivate	computeserver-0-0
13	CAMERAHOSTS	ComputeServerPrivate	computeserver-test-0-0
14	CAMERAHOSTS	ComputeServerPrivate	computeserver-test-0-0
0	CAMERAHOSTS	ForumsServer	web7.camera.calit2.net
0	CAMERAHOSTS	GamaServer	gama-camera.rocksclusters.org
0	CAMERAHOSTS	GridSphereServer	web7.camera.calit2.net
0	CAMERAHOSTS	PostgresServerPrivate	postgres8db-0-3
13	CAMERAHOSTS	PostgresServerPrivate	postgres8db-0-0
14	CAMERAHOSTS	PostgresServerPrivate	postgres8db-0-0
0	CAMERAHOSTS	PRWebServer	camera.calit2.net
0	CAMERAHOSTS	SSOServer	web6.camera.calit2.net
0	CAMERAHOSTS	SSOServerPrivate	sso-0-0.local

13 == Appserver Testing Appliance

14 == Computeserver Testing Appliance



# Summary of Layout

---

- ◆ Different servers must be applied as a group
  - [application, camera compute, database, SSO]
  - We have production and staging (test) versions
- ◆ Configuration variables are held in a database
  - Data “pointers” are supported using <indirect>
  - Rocks Kickstart framework dereferences all <indirect> statements
- ◆ Today, there are 142 direct and indirect references in our Configuration Database
  - 72 of these are <indirects>



# Viz & CAMERA Taught Us

---

- ◆ Viz and CAMERA helped Rocks
  - ⇒ Better abstractions in XML
  - ⇒ Learned new ways to use the existing framework
- ◆ We were surprised at
  - ⇒ How small the differences are between Viz, CAMERA, and HPC Clusters
  - ⇒ How well our framework supported such functionally different systems