Overview of Rolls

Rocks-A-Palooza I

Track 2

Session 1

Rocks Philosophy

- We’ve developed a “cluster compiler”
  - XML framework + XML parser + kickstart file generator
  - Source code + preprocessor + linker
- Think about “programming your cluster”
  - Not “administering your cluster”

Rocks Philosophy

- Rocks is the "syringe" for software deployment
- You determine the function of the cluster by selecting Rolls
- Then you "push" the deployment to all nodes

Goal of Rolls

- Develop a method to reliably install software on a frontend
- “User-customizable” frontends
- Two established approaches:
  - Add-on method
  - Rocks method

Add-on Method

1. User responsible for installing and configuring base software stack on a frontend
2. After the frontend installation, the user downloads ‘add-on’ packages
3. User installs and configures add-on packages
4. User installs compute nodes

Major issue with add-on method

- The state of the frontend before the add-on packages are added/configured is unknown

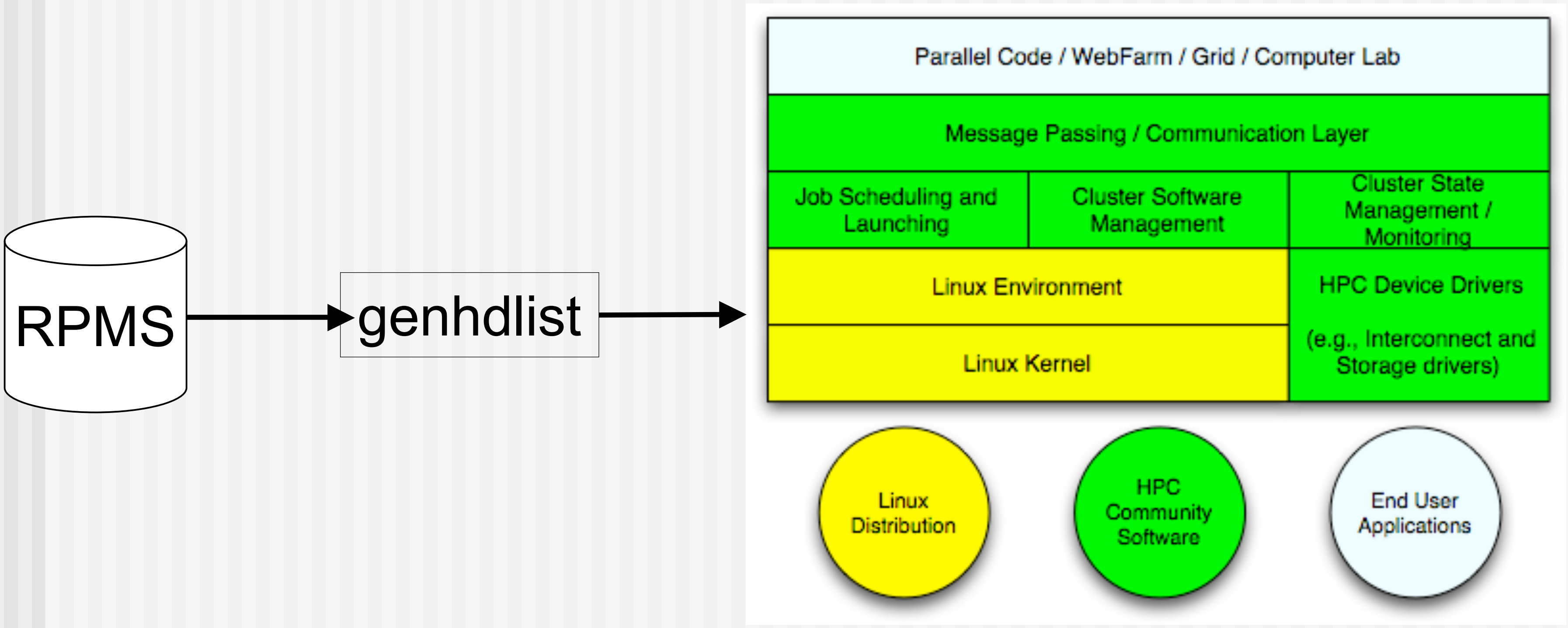
Rocks Method

- To address the major problem with the add-on method, we had the following idea:
  - All non-RedHat packages must be installed and configured in a controlled environment
- A controlled environment has a known state
- We chose the RedHat installation environment for the controlled environment

Goal of Rolls

- This led to modifying the standard RedHat installer in order to accept new packages and configuration
- A tricky proposition
  - A RedHat distribution is a monolithic entity
    - It's tightly-coupled
    - A program called "genhdlist" creates a binary files (hdlist and hdlist2) that contain metadata about every RPM in the distribution
- To add/remove/change an RPM, you need to re-run genhdlist
  - Else, the RedHat install will not recognize the package
  - Or worse, it fails during package installation

Monolithic Software Stack



RPMS

genhdlist

Parallel Code / WebFarm / Grid / Computer Lab

Message Passing / Communication Layer

Job Scheduling and Launching

Cluster Software Management

Cluster State Management / Monitoring

Linux Environment

Linux Kernel

HPC Device Drivers

(e.g., Interconnect and Storage drivers)

Linux Distribution

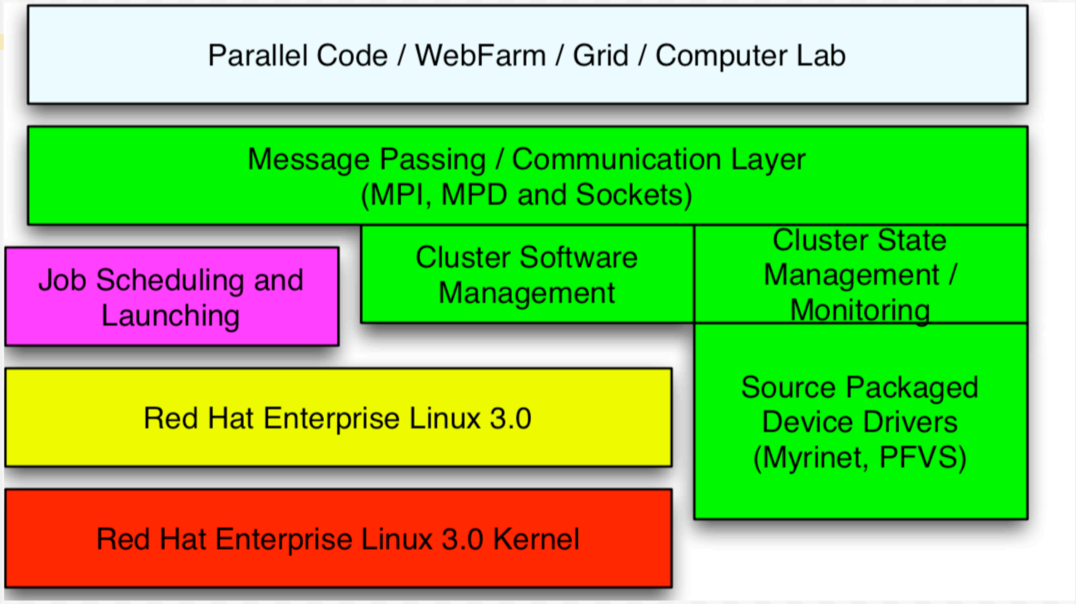
HPC Community Software

End User Applications

Goal of Rolls

- Problem: To make the frontend user-customizable at installation time, we needed a mechanism that could accept new packages
- And, we still wanted to leverage the RedHat installer
  - We don’t want to be in the installer business
- Solution: Our implementation makes the RedHat installer “think” it is just installing a monolithic RedHat distribution

Goal of Rolls



Parallel Code / WebFarm / Grid / Computer Lab

Message Passing / Communication Layer (MPI, MPD and Sockets)

Job Scheduling and Launching

Cluster Software Management

Cluster State Management / Monitoring

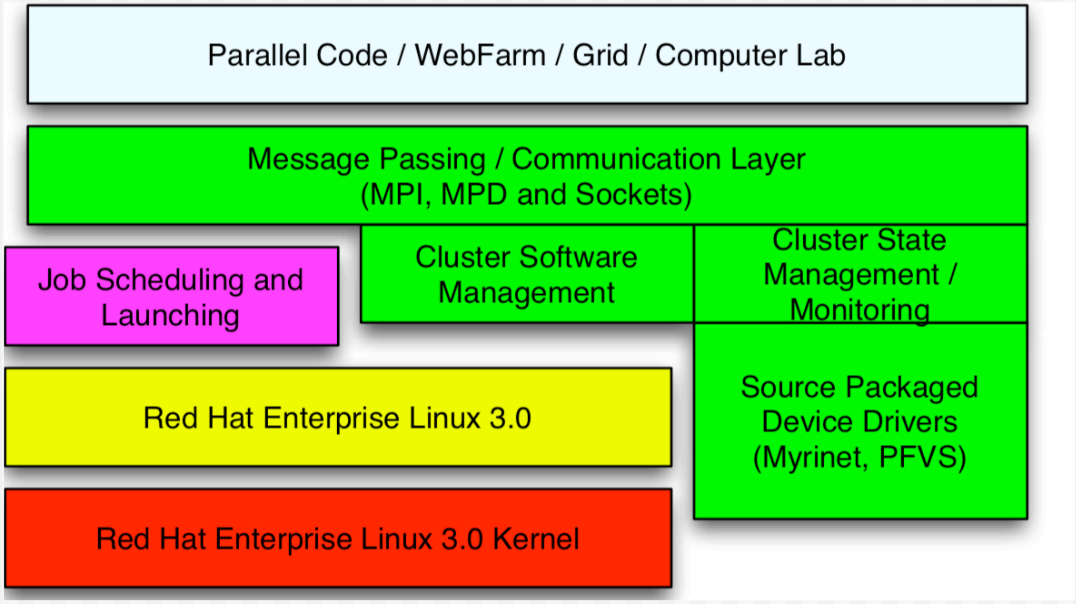
Red Hat Enterprise Linux 3.0

Source Packaged Device Drivers (Myrinet, PFVS)

Red Hat Enterprise Linux 3.0 Kernel

- How do you make all the packages above look like a monolithic distribution?
  - Easy! Just run “genhdlist” at release time!
- But, how do you do it when some of the above blocks are optional and/or unknown?
  - An “unknown” block is one produced after the release or by a third-party

Rolls Function and Value


Parallel Code / WebFarm / Grid / Computer Lab
Message Passing / Communication Layer (MPI, MPD and Sockets)
Job Scheduling and Launching
Cluster Software Management
Cluster State Management / Monitoring
Red Hat Enterprise Linux 3.0
Source Packaged Device Drivers (Myrinet, PFVS)
Red Hat Enterprise Linux 3.0 Kernel

- Function: Rolls extend/modify stock RedHat
- Value: Third parties can extend/modify Rocks
  - Because Rolls can be optional

The RedHat Installer

Anaconda: RedHat’s Installer

- Open-source python-based installer
- Developed by RedHat
- (Somewhat) object-oriented
  - We extend when we can and insert “shims” when we can’t

Anaconda: RedHat’s Installer

- Key tasks:
  - Probe hardware
  - Ask users for site-specific values
    - E.g., IP addresses and passwords
  - Insert network and storage drivers
    - For network-based installations and to write packages down onto local disk
  - Install packages
    - RPMs
  - Configure services
    - Via shell scripts

Scripted Installation

- Anaconda achieves “lights-out” installation via kickstart mechanism
- It reads a “kickstart file”
  - Description of how to install a node
- One file composed of three key sections:
  - Main: general parameters
  - Packages: list of RPMs to install
  - Post: scripts to configure services

Kickstart File

- Main section

```
rootpw --iscrypted oijewfl5853812kfd
url --url http://10.1.1.1/install/rocks-dist/lan/enterprise/3/en/os/i386
zerombr yes
bootloader --location=mbr
lang en_US
langsupport --default en_US
keyboard us
mouse genericps/2
text
install
reboot
timezone --utc America/Los_Angeles
part
```

Kickstart File

- Packages section

```
%packages --ignoredeps --ignoremissing
@Base
PyXML
atlas
autofs
bc
chkrootkit
contrib-pexpect
contrib-pvfs-config
contrib-python-openssl
```

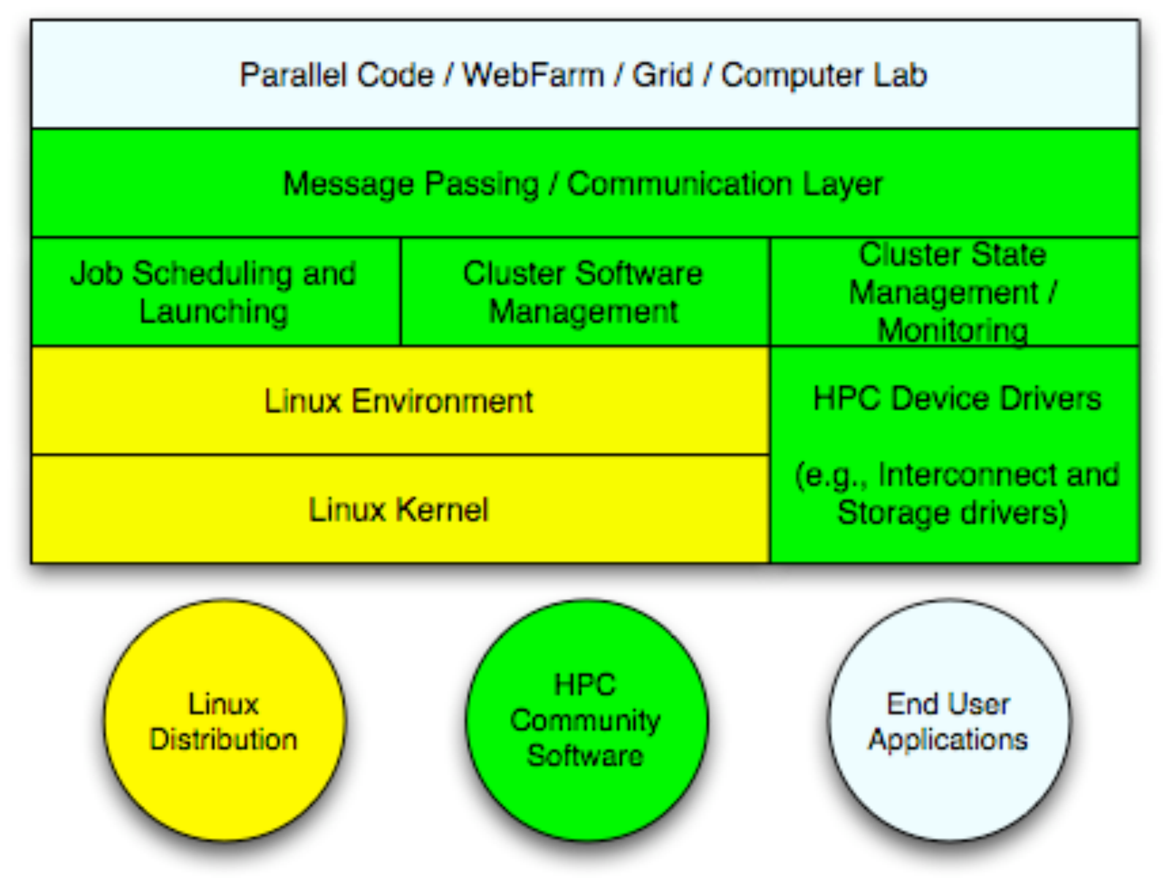
Kickstart File

- Post section

```
%post

cat > /etc/motd << 'EOF'
Rocks Compute Node
EOF
```

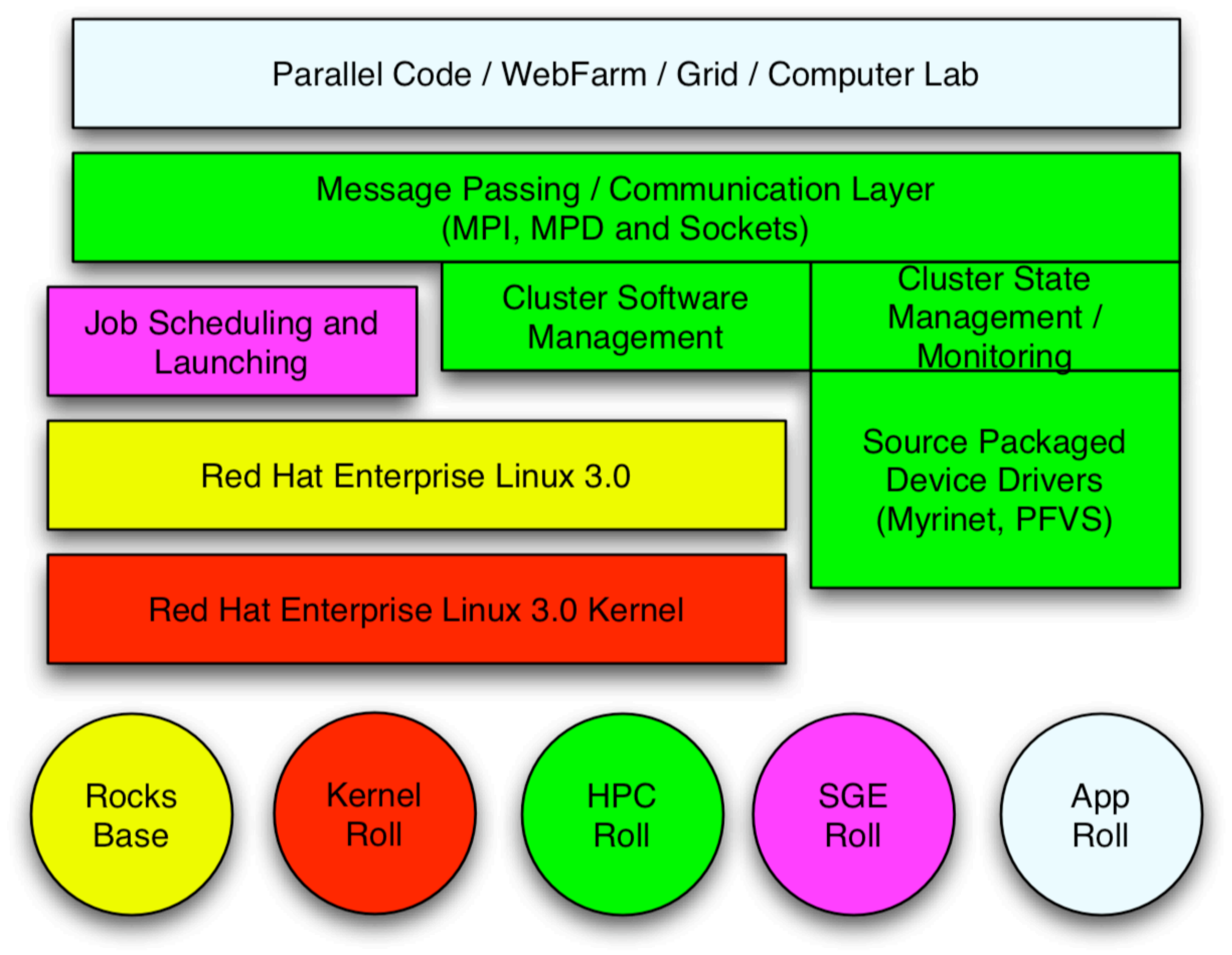
Rolls High-Level Description

Monolithic Software Stack



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

Rolls



Parallel Code / WebFarm / Grid / Computer Lab

Message Passing / Communication Layer (MPI, MPD and Sockets)

Job Scheduling and Launching

Cluster Software Management

Cluster State Management / Monitoring

Red Hat Enterprise Linux 3.0

Source Packaged Device Drivers (Myrinet, PFVS)

Red Hat Enterprise Linux 3.0 Kernel

Rocks Base

Kernel Roll

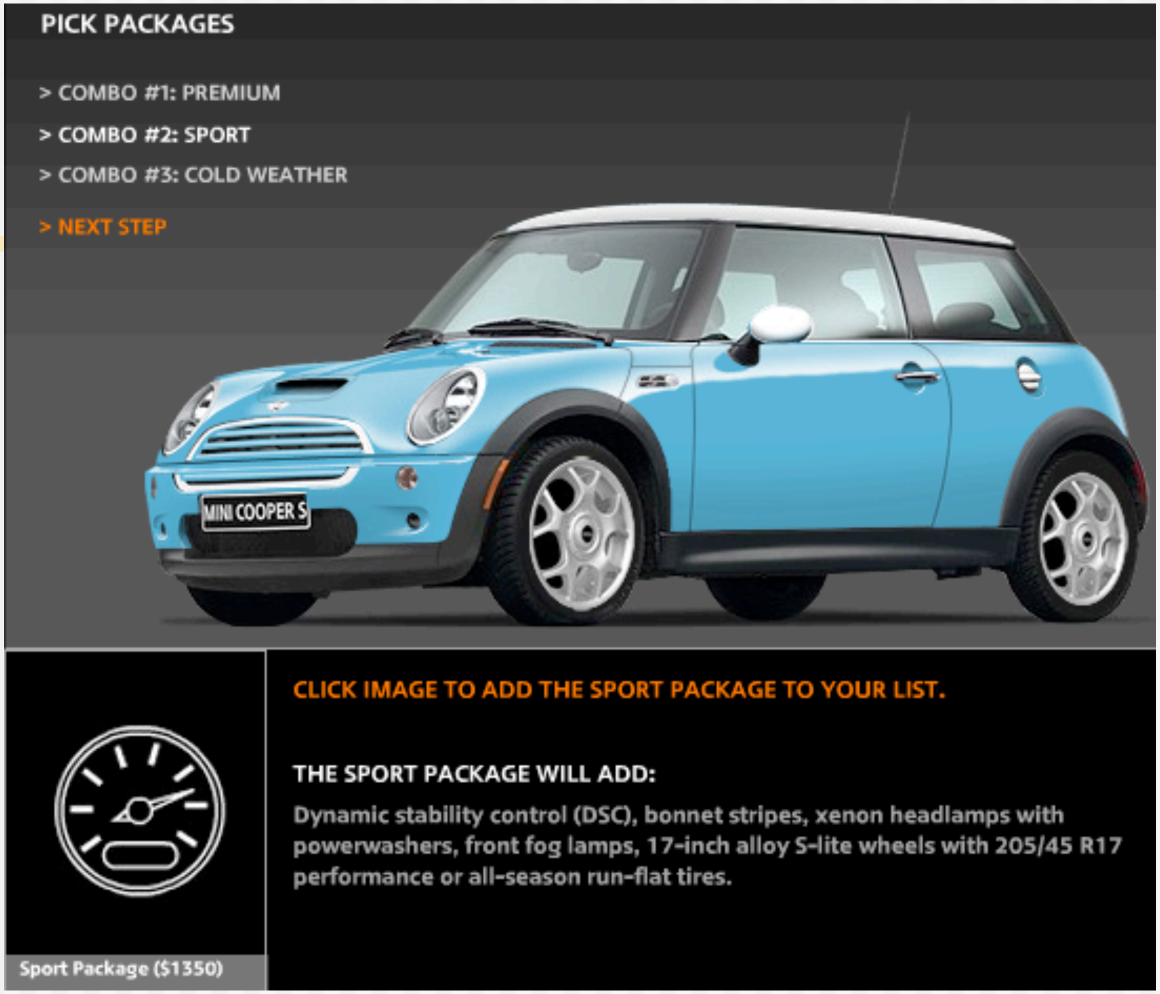
HPC Roll

SGE Roll

App Roll

- Dissecting the monolithic software stack

Rolls



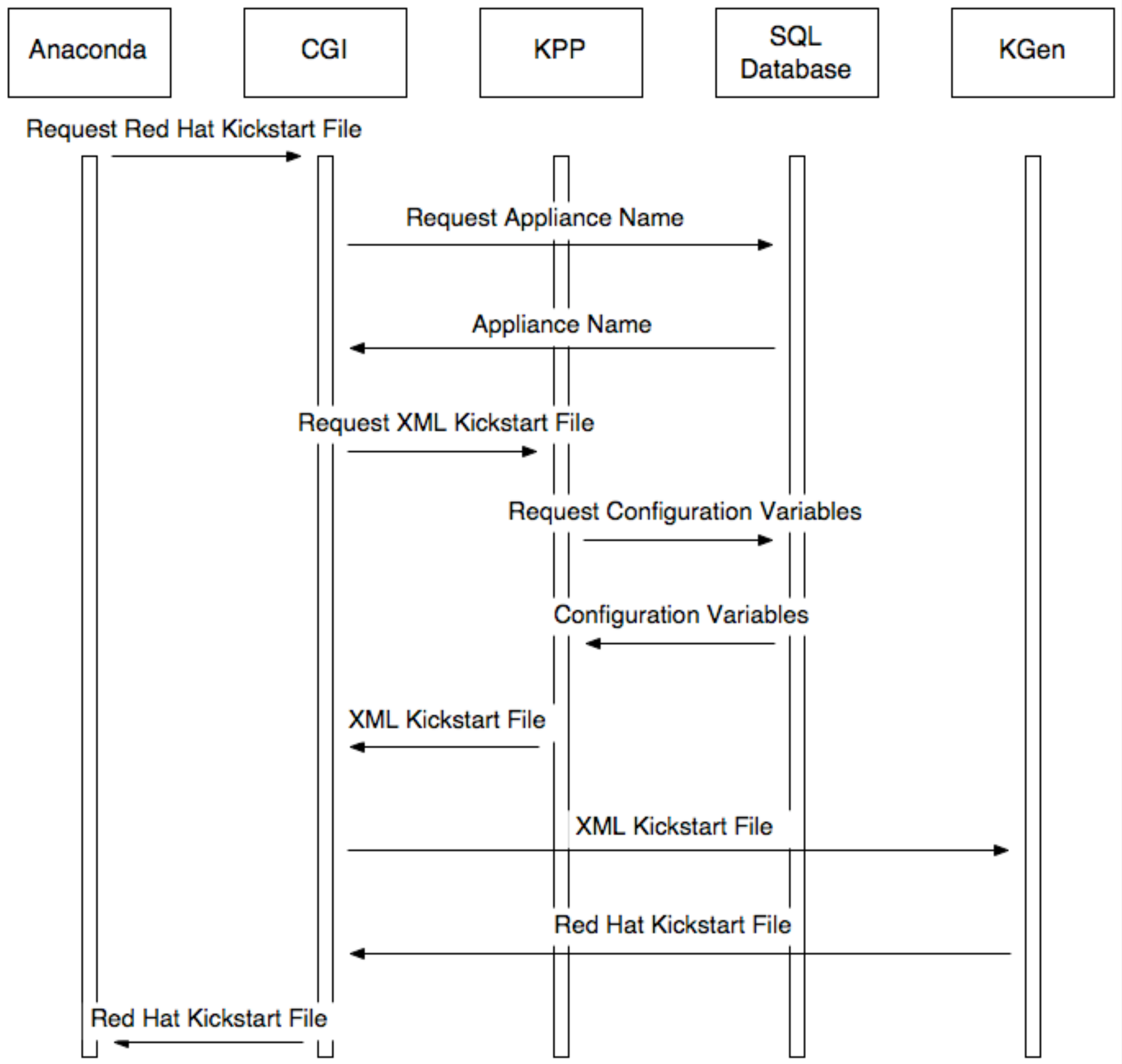
CLICK IMAGE TO ADD THE SPORT PACKAGE TO YOUR LIST.

THE SPORT PACKAGE WILL ADD:

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

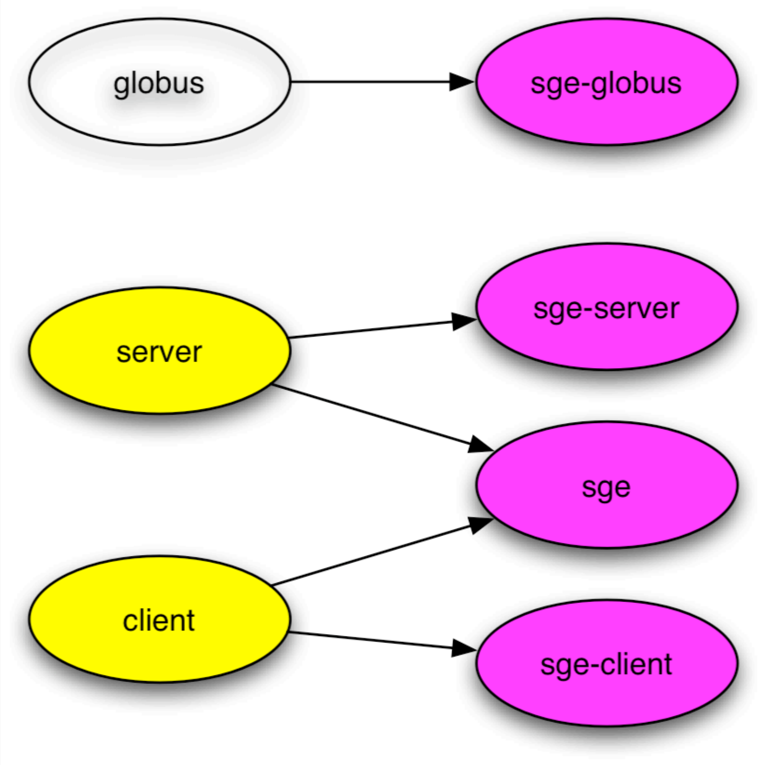
- Think of a roll as a “package” on a car

Getting A Kickstart File



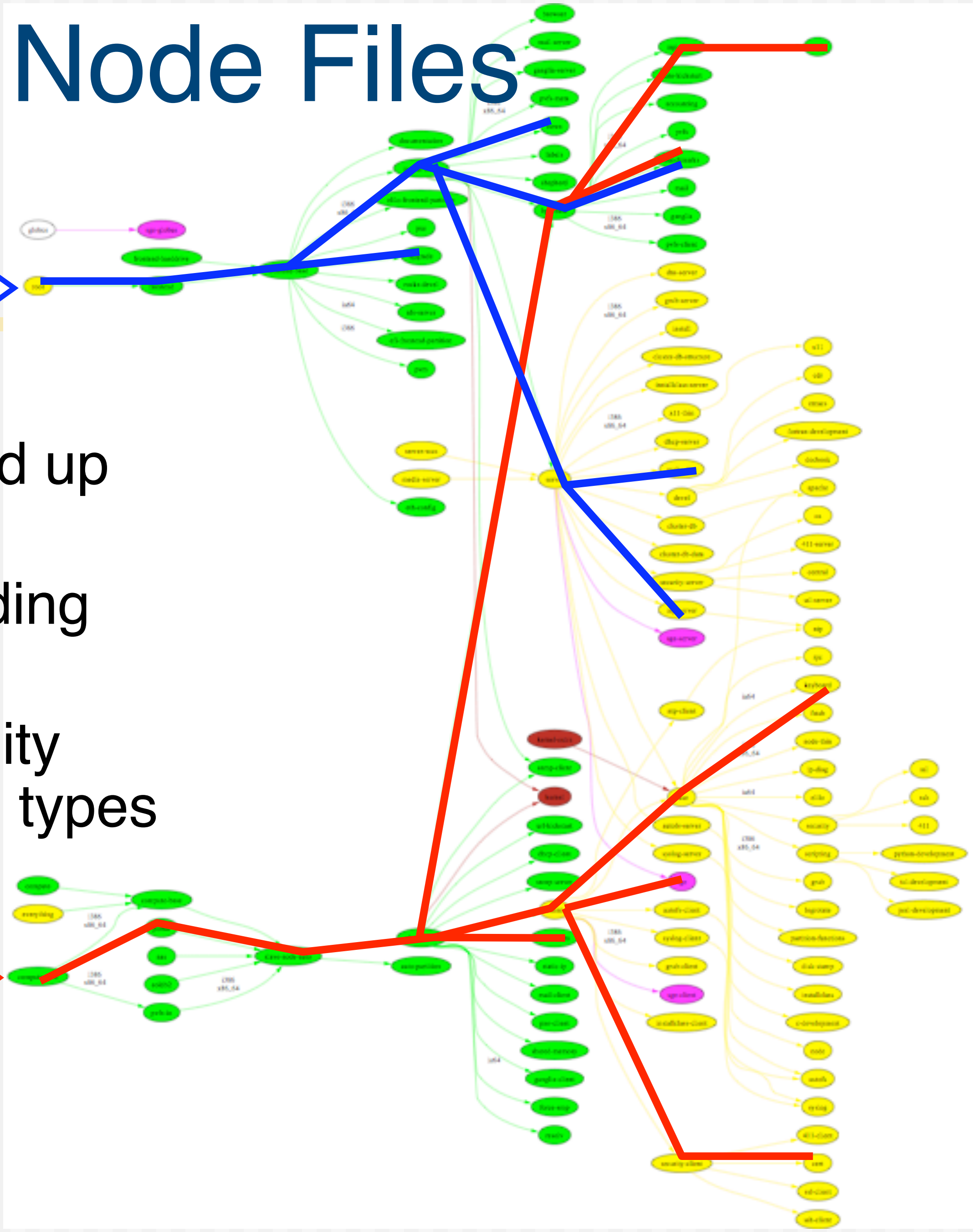
Use Graph Structure to Dissect Distribution

- Use ‘nodes’ and ‘edges’ to build a customized kickstart file
- Nodes contain portion of kickstart file
  - Can have a ‘main’, ‘package’ and ‘post’ section in node file
- Edges used to coalesce node files into one kickstart file



Coalescing Node Files

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

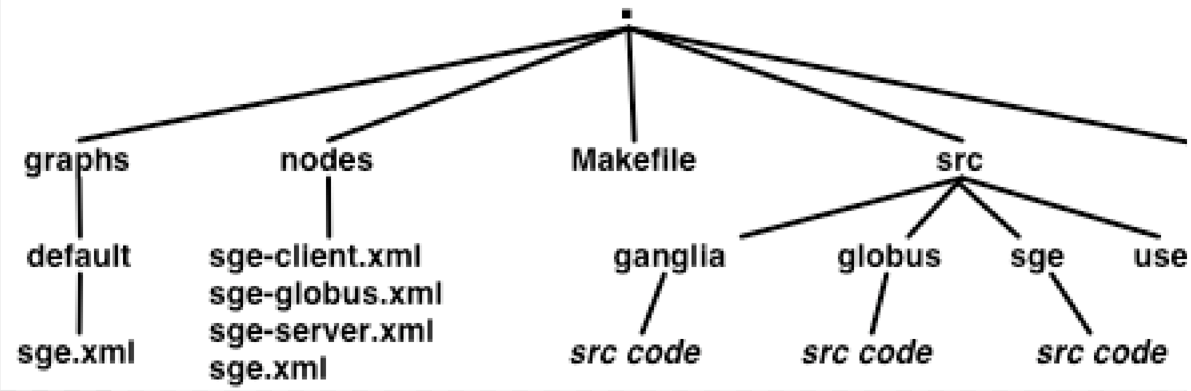


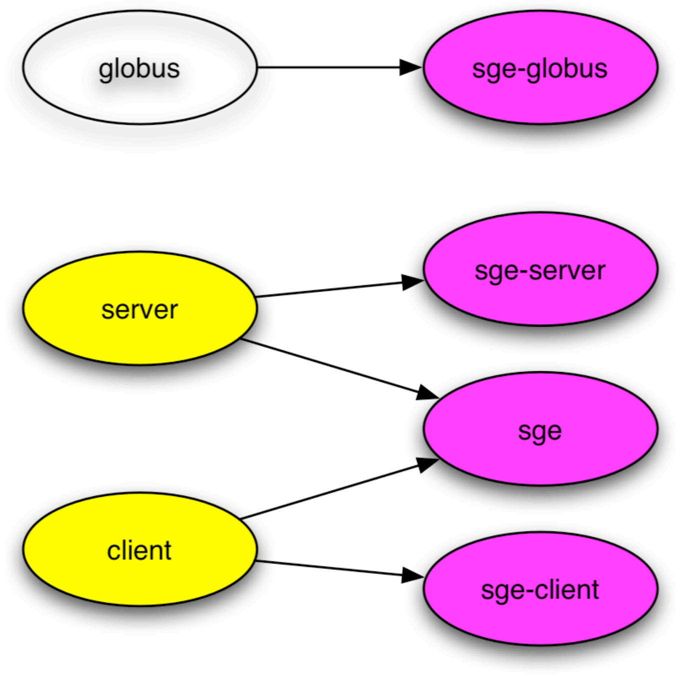
Frontend Root

Compute Root

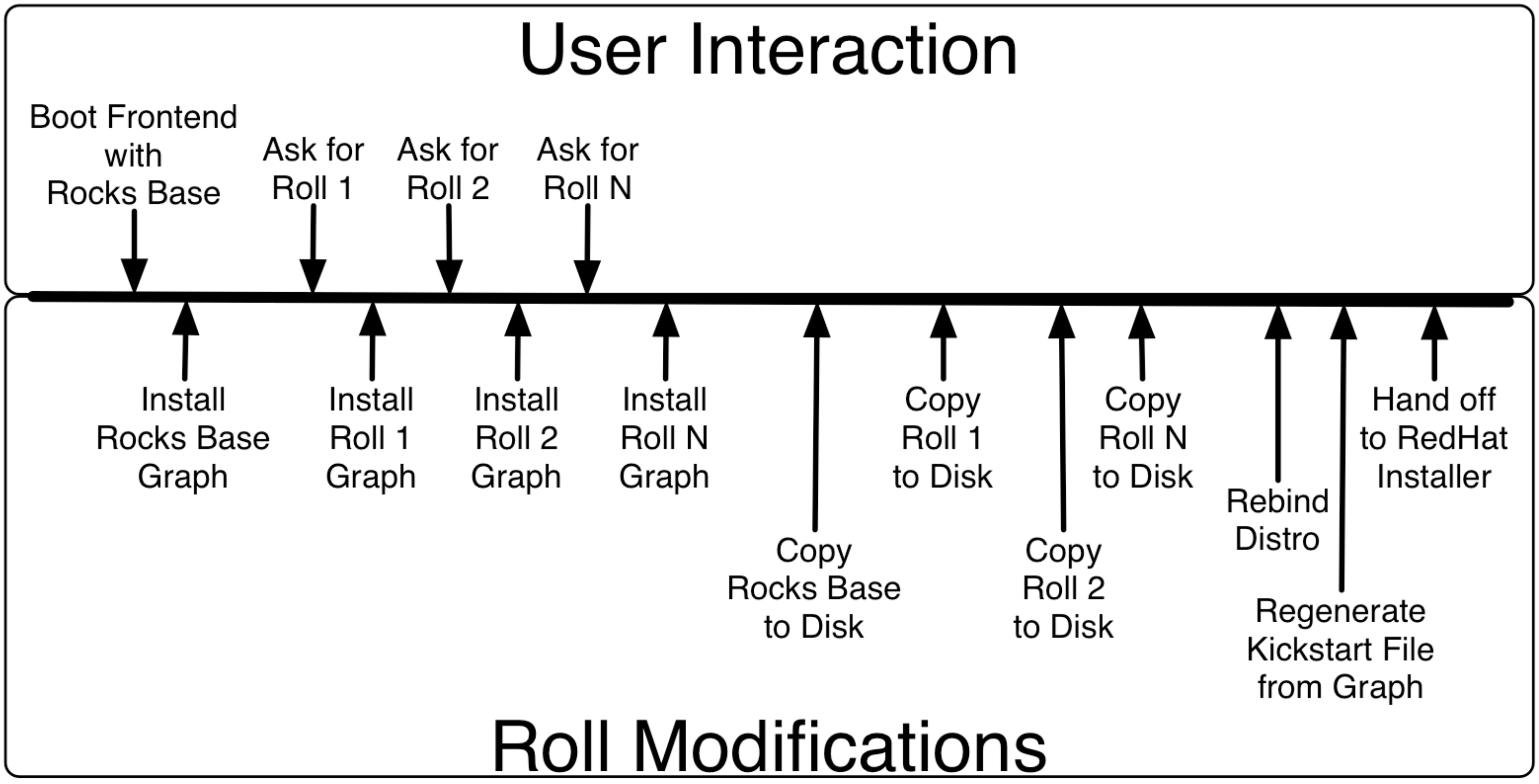
Why We Use A Graph

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

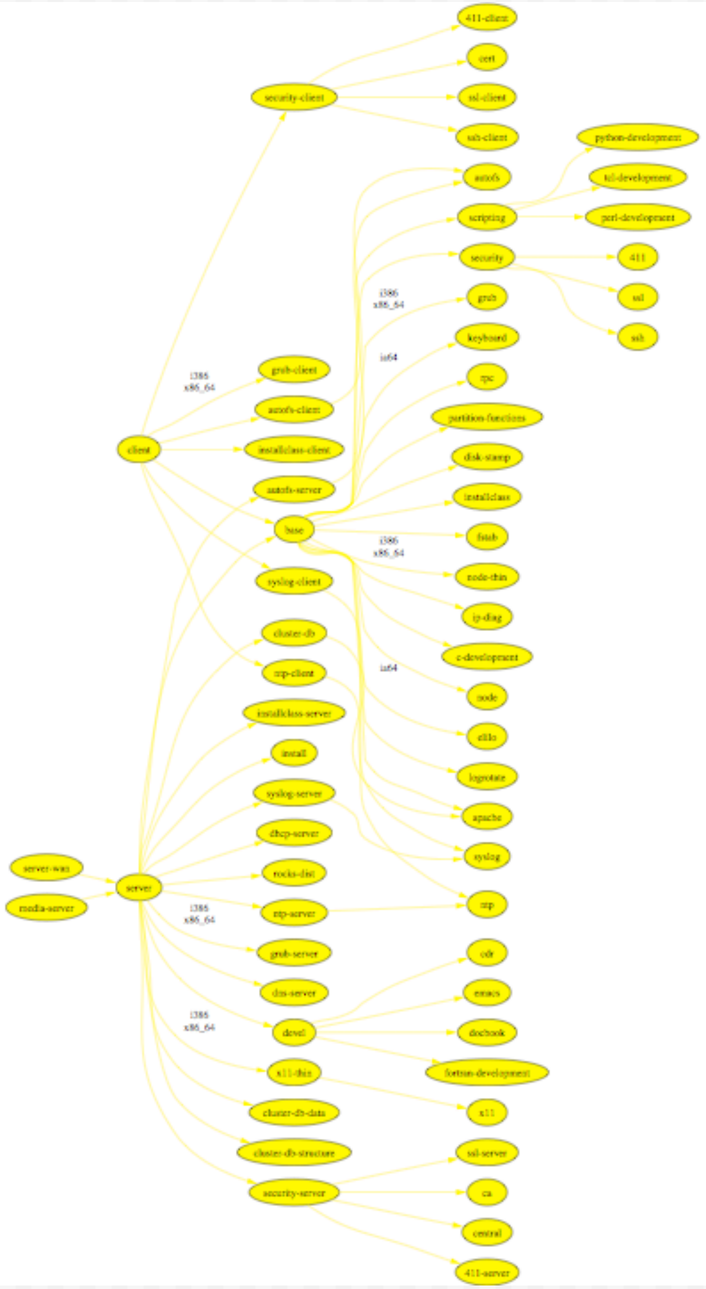




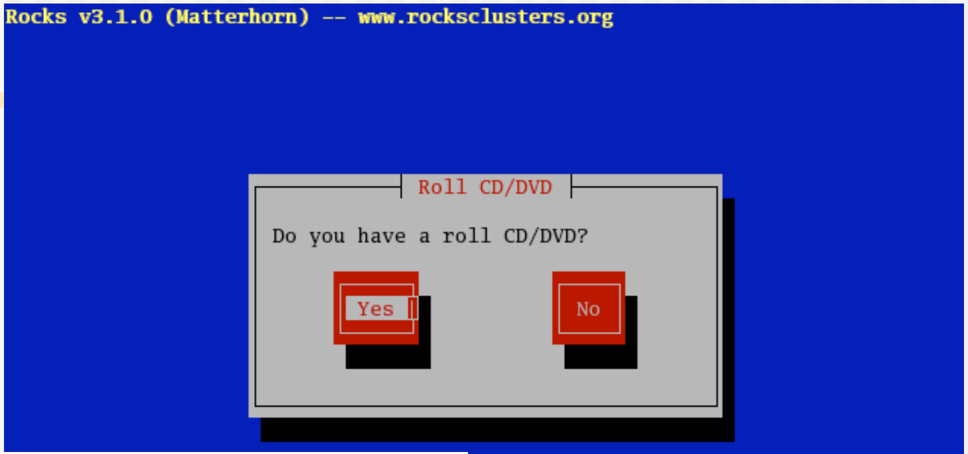
Rocks Extensions Installation Timeline



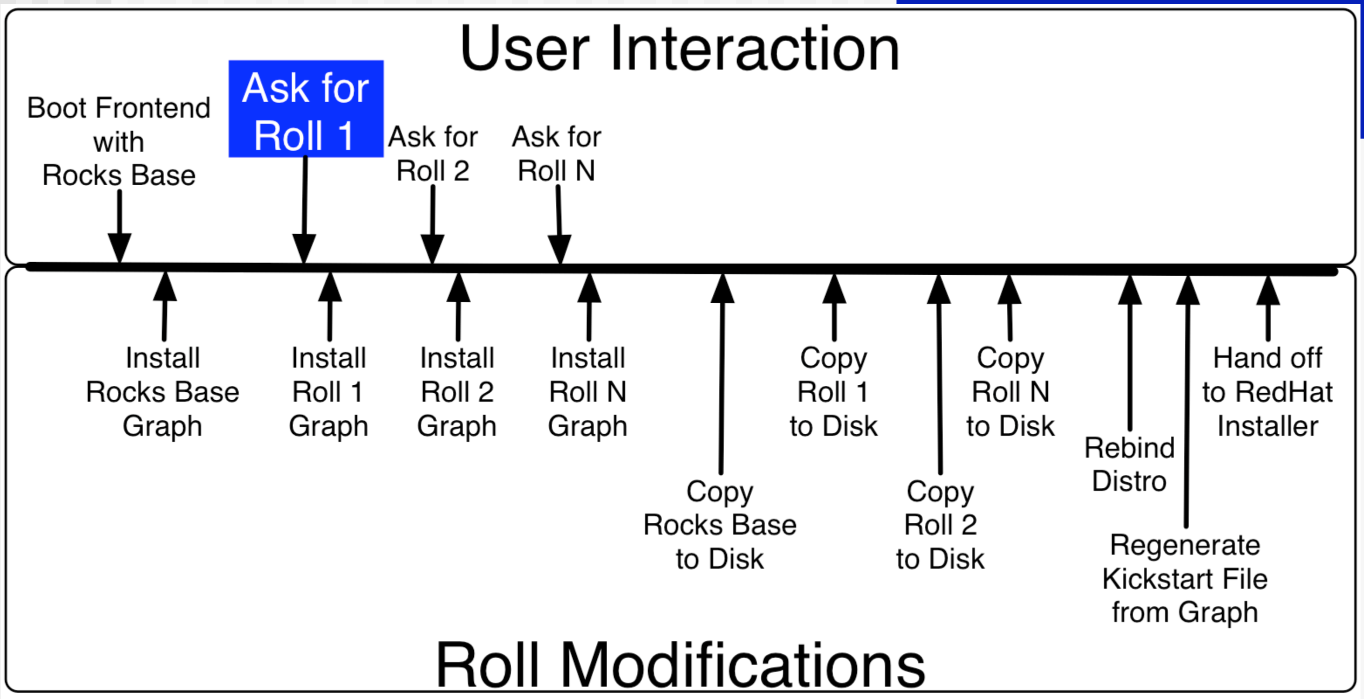
Install Rocks Base Graph



Anaconda Modified to Accept Rolls



```
Rocks v3.1.0 (Matterhorn) -- www.rocksclusters.org

                Roll CD/DVD
    Do you have a roll CD/DVD?

        Yes              No

<Space> selects   |   <F12> next screen
```



User Interaction

Boot Frontend with Rocks Base

Ask for Roll 1

Ask for Roll 2

Ask for Roll N

Install Rocks Base Graph

Install Roll 1 Graph

Install Roll 2 Graph

Install Roll N Graph

Copy Rocks Base to Disk

Copy Roll 1 to Disk

Copy Roll 2 to Disk

Copy Roll N to Disk

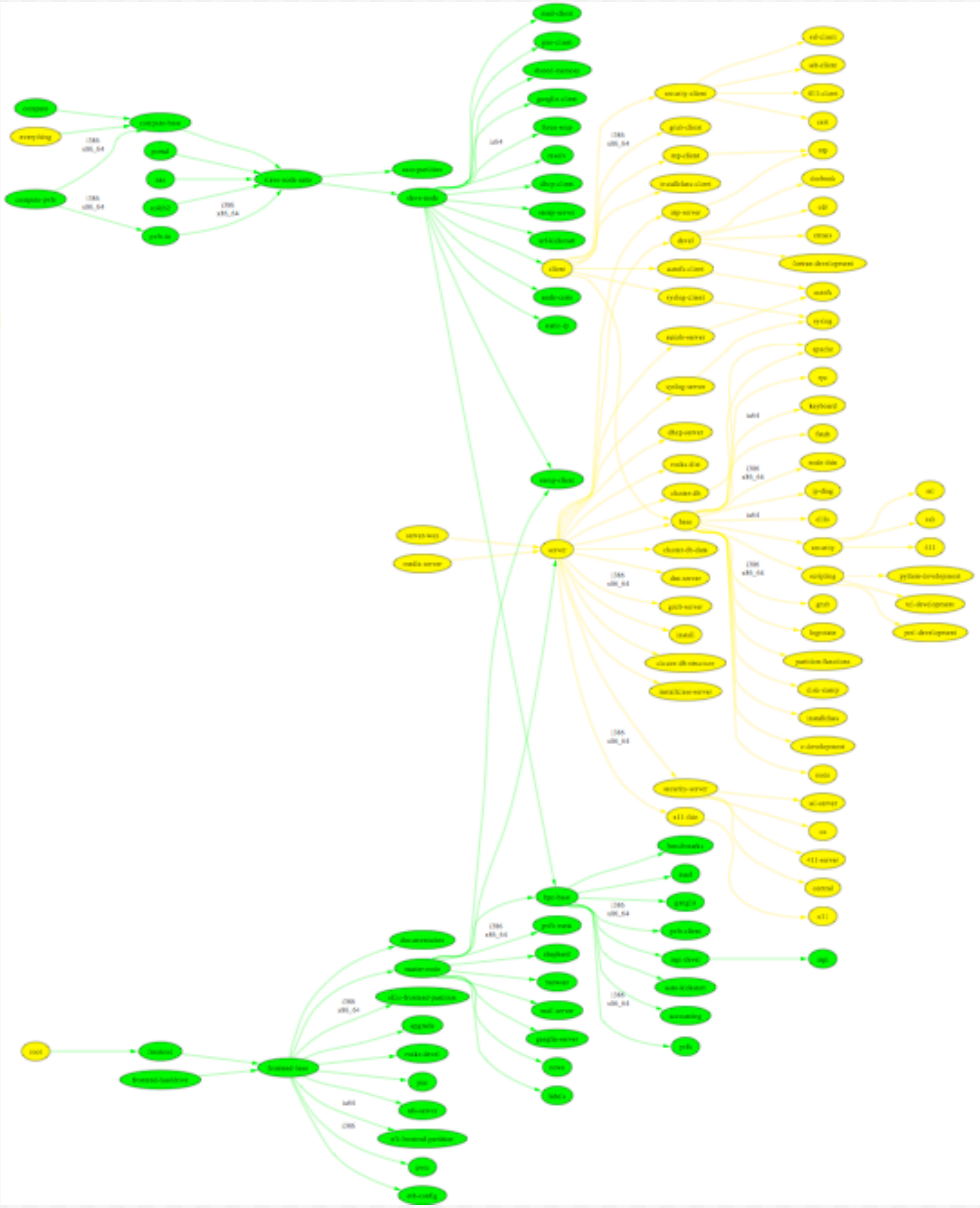
Rebind Distro

Regenerate Kickstart File from Graph

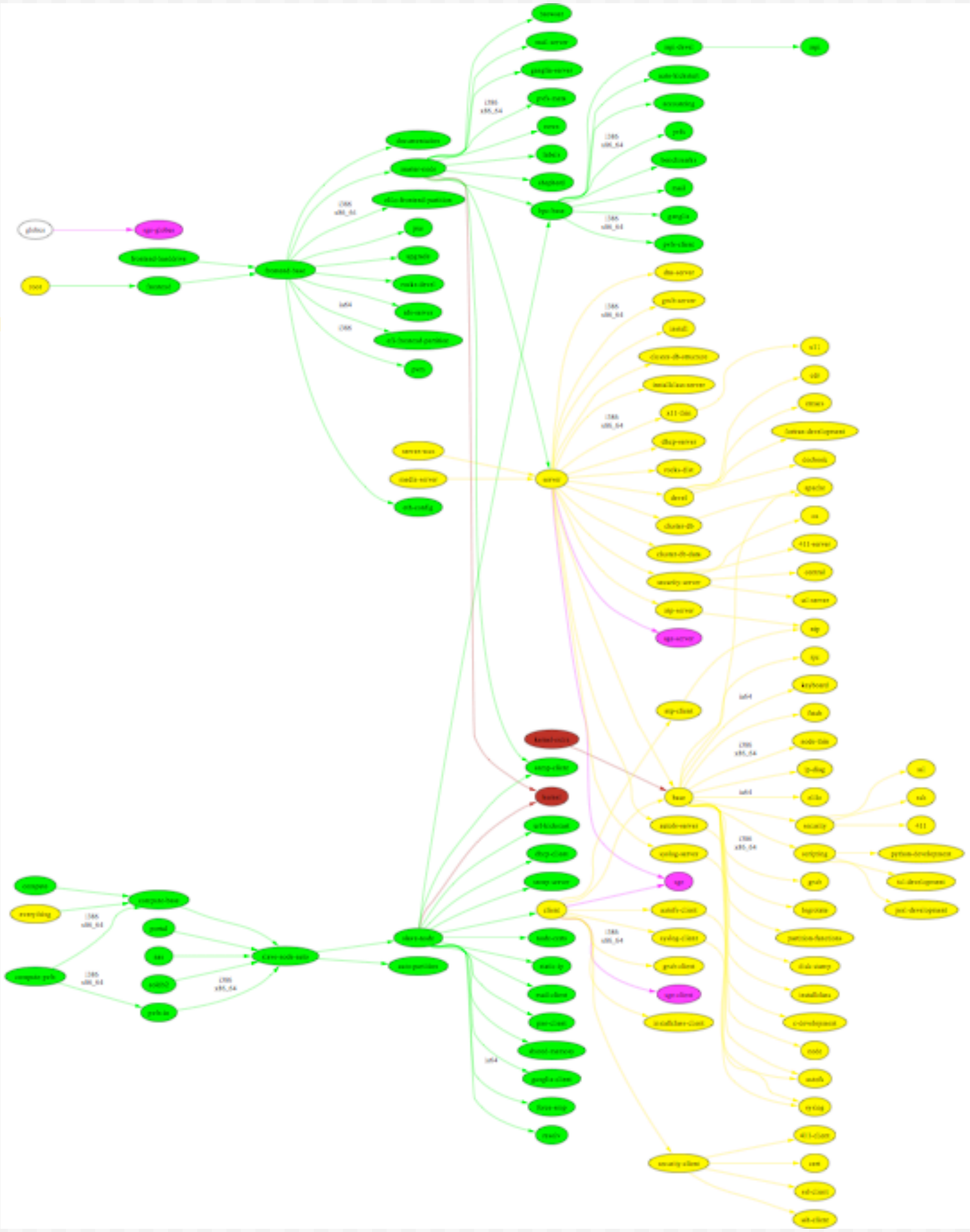
Hand off to RedHat Installer

Roll Modifications

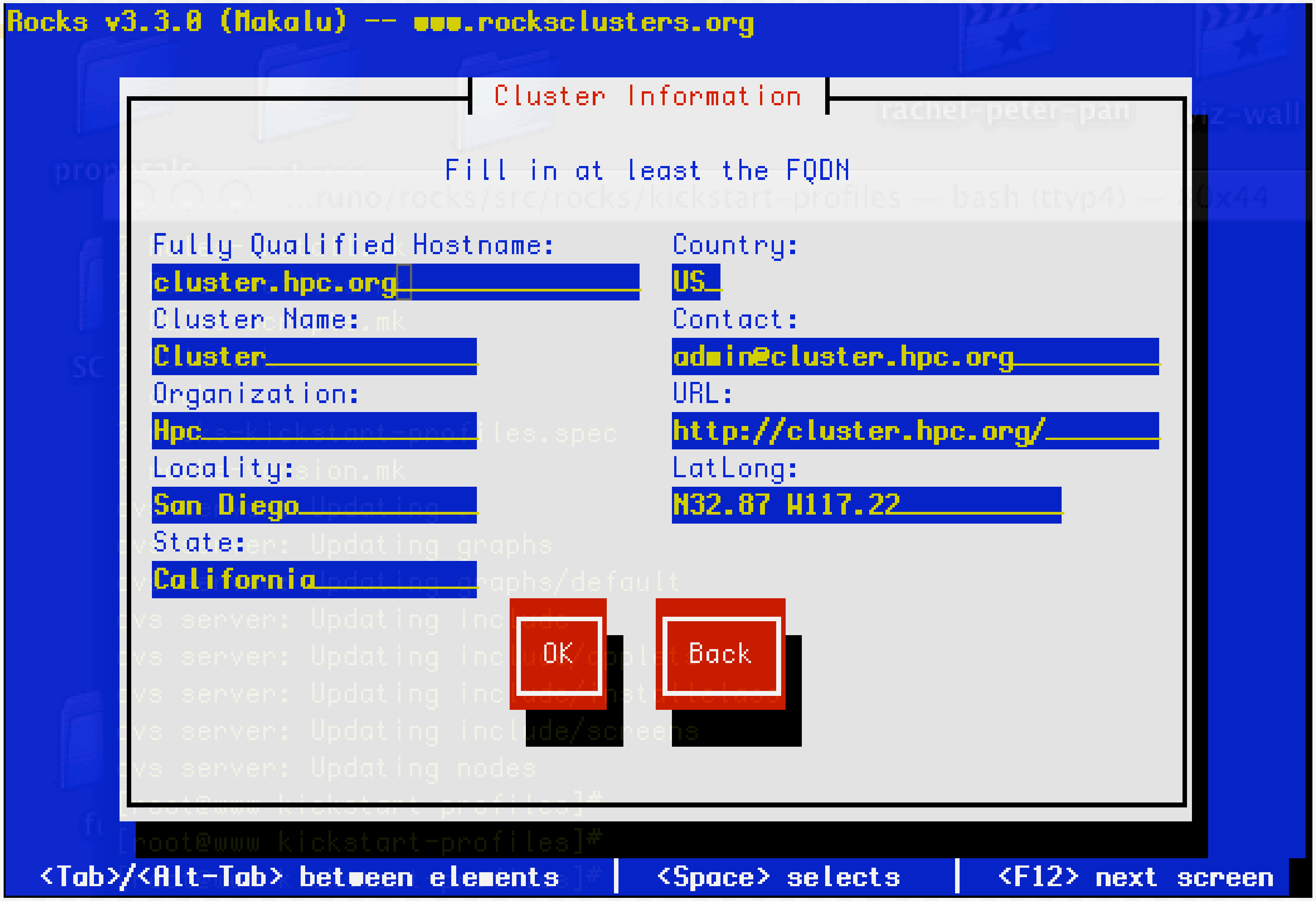
Install Roll Graph



Base + All Rolls



Anaconda Modified to Display New User Input Screens



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

Cluster Information

Fill in at least the FQDN

Fully Qualified Hostname: cluster.hpc.org

Cluster Name: Cluster

Organization: Hpc

Locality: San Diego

State: California

Country: US

Contact: admin@cluster.hpc.org

URL: http://cluster.hpc.org/

LatLong: N32.87 W117.22

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

Anaconda Modified to Display New User Input Screens

- How we do it:
  - Place a shim in Anaconda to call our screens instead of the ‘betanag’ RedHat screen

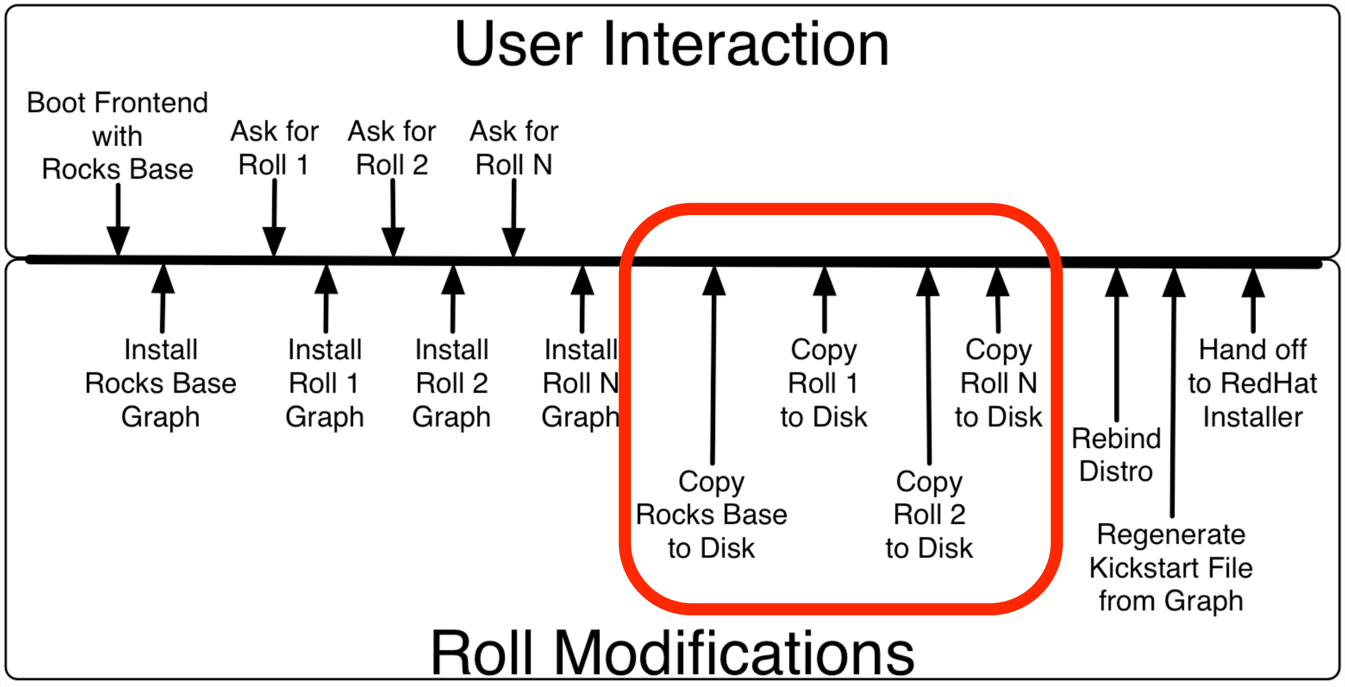
```
index = 0
for key in installSteps:
        if key[0] == "betanag":
                break
        index = index + 1

installSteps[index] = ("rockswindows", ("id.rocks", ))

# set list of user-defined windows
dispatch.skipStep("rockswindows", skip = 0)
stepToClasses["rockswindows"] = ("ksclass",
        tuple(rockswindows))
```

Anaconda Modified to Display New User Input Screens

- How you use it:
  - In your XML file:

```
<installclass>

class CACLInfoWindow:

def __call__(self, screen, Info):

                bb = ButtonBar (screen, (TEXT_OK_BUTTON, ("Back","back")))
                toplevel = GridFormHelp (screen,
                        _("CACL Setup Information"), "CACLInfo", 1, 3)
                leftGrid = Grid(1,12)
                rightGrid = Grid(1,12)
                infoGrid = Grid(2,1)
                .
                .

addScreen("CACLInfoWindow")

</installclass>
```

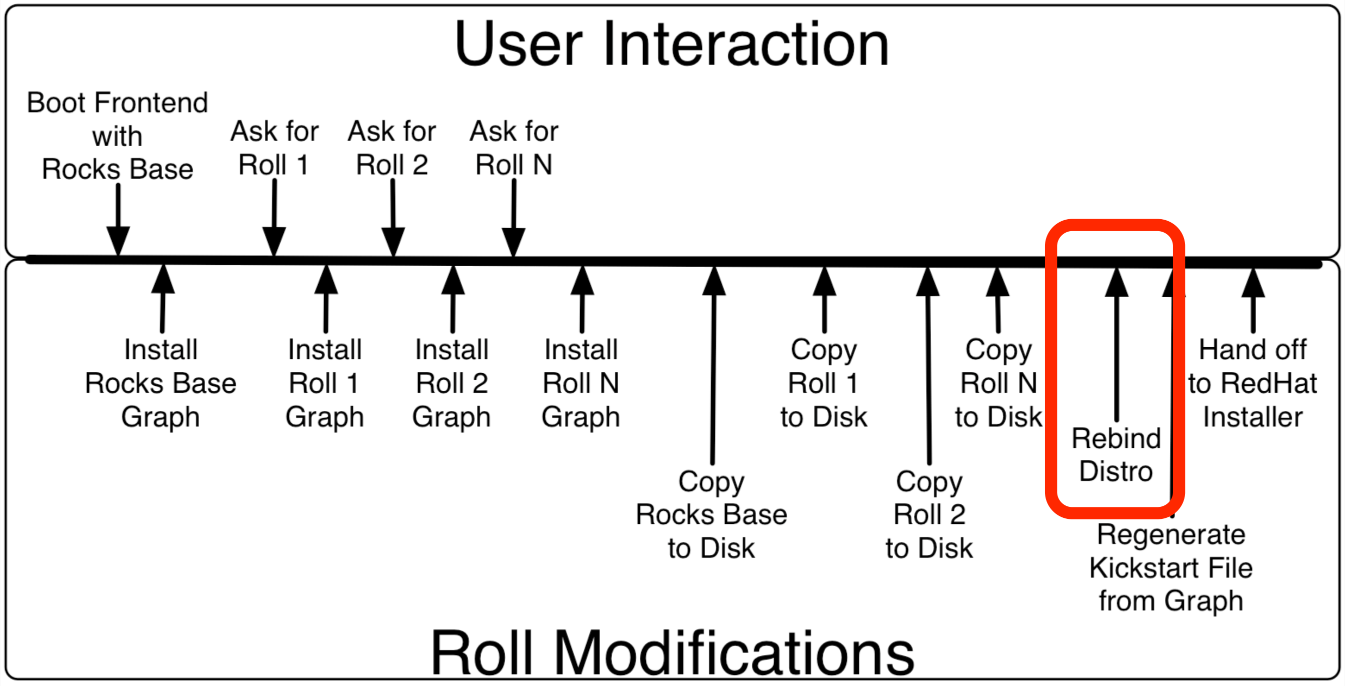
Copy Media To Local Disk



- Base and all user-supplied Rolls are copied to local disk
  - These packages are used to install compute nodes

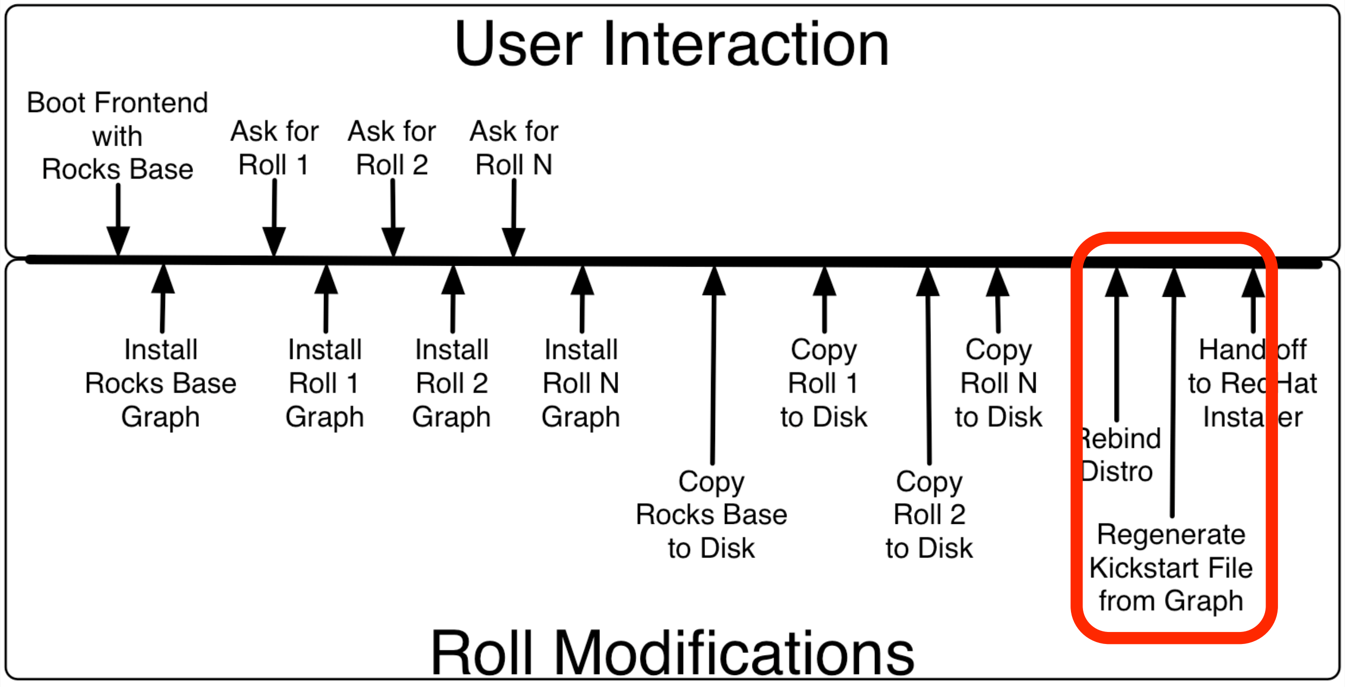
© 2005 UC Regents

Rebind Distro



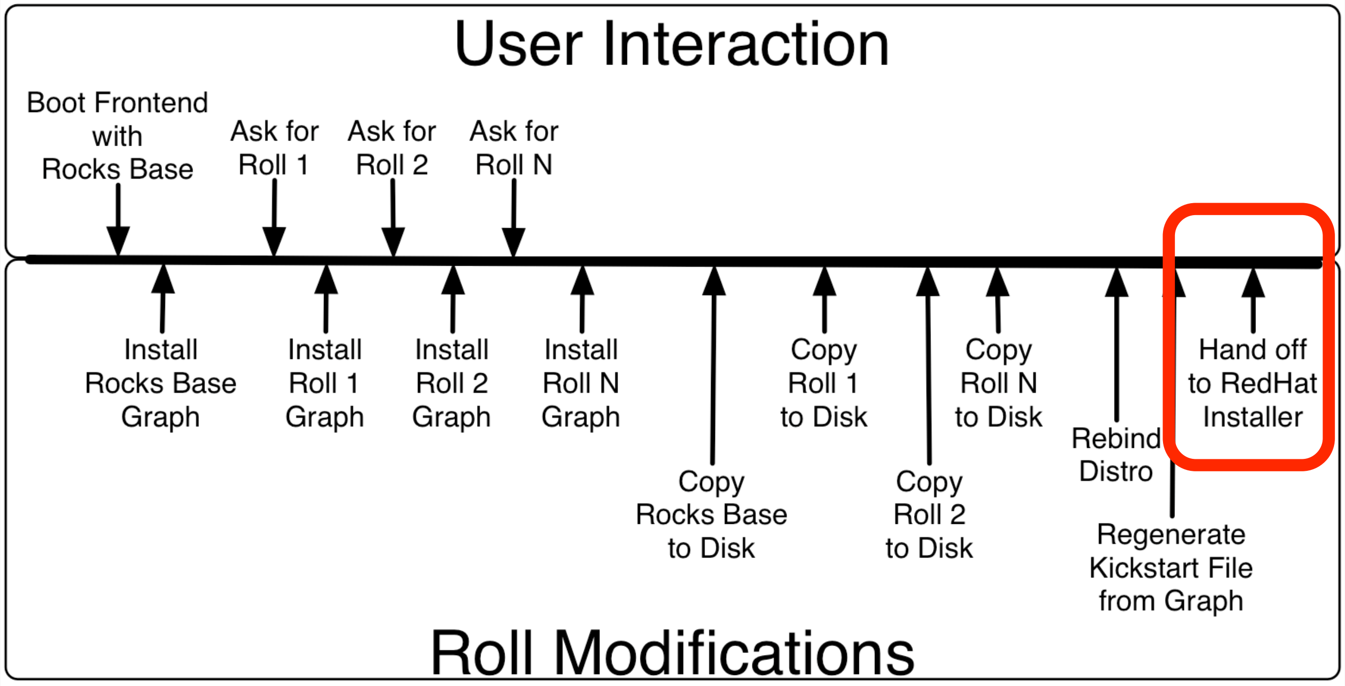
- Merge base with rolls into one RedHat-compliant distribution
  - This takes the dissected distro and tightly binds it
    - Note: We actually install the frontend off the local hard disk (not the CD media)

Rebuild the Kickstart File



- Traverse the final graph using the node ‘frontend’ as the root
  - Allows us to customize a frontend configuration at install time

Hand Off To RedHat

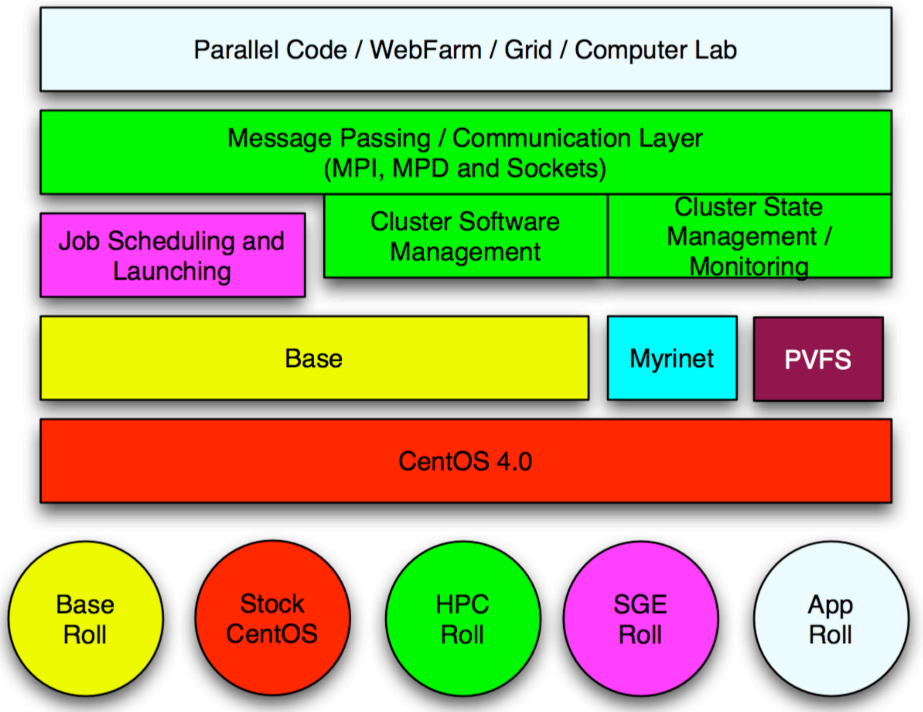


- Anaconda has no idea what hit it!
- The remainder of the installation looks like a standard RedHat installation (just with more packages and cluster-specific configuration)

Near Future

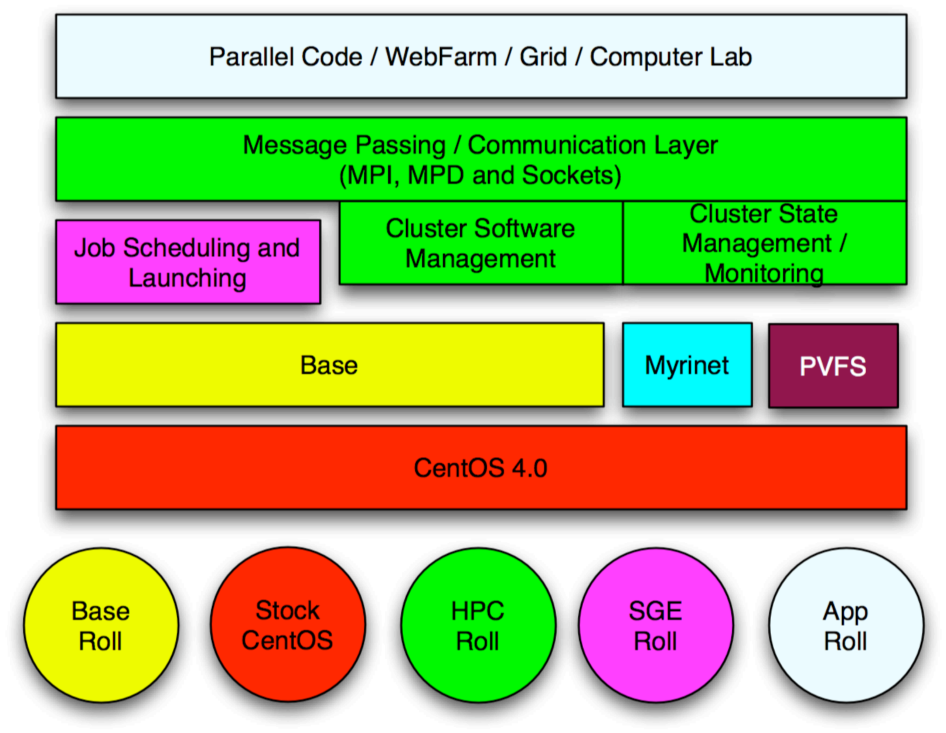
Rocks 4.0.0

- Everything is a Roll!


Parallel Code / WebFarm / Grid / Computer Lab
Message Passing / Communication Layer (MPI, MPD and Sockets)
Job Scheduling and Launching
Cluster Software Management
Cluster State Management / Monitoring
Base
Myrinet
PVFS
CentOS 4.0
Base Roll
Stock CentOS
HPC Roll
SGE Roll
App Roll

- Using CentOS 4.0 as base OS
  - People have also successfully substituted Scientific Linux and RHEL 4 for CentOS
- When asked for roll, input stock CentOS CDs

Rocks 4.0.0


Parallel Code / WebFarm / Grid / Computer Lab
Message Passing / Communication Layer (MPI, MPD and Sockets)
Job Scheduling and Launching
Cluster Software Management
Cluster State Management / Monitoring
Base
Myrinet
PVFS
CentOS 4.0
Base Roll
Stock CentOS
HPC Roll
SGE Roll
App Roll

- Dissecting existing Rolls
  - HPC roll is much smaller
    - E.g., Myrinet and PVFS are separate Rolls

Loader Modifications

Loader Modifications

- The first program that runs during a RedHat install is a C program called “loader”
- Performs low-level setup
  - Loads drivers
  - Configures network
  - Downloads anaconda
  - Gets kickstart file

Loader Modifications

- Make HTTP the default install method
  - RedHat uses NFS as default
- Rationale
  - Installation is read-only, don’t need a file system
  - HTTP traffic can be easily load balanced
  - In terms of active development, more people are working on HTTP than NFS

Loader Modifications

- Robust kickstart file acquistion
  - 10 retries to get kickstart file
    - RedHat has only 1
  - NACK to throttle kickstart file acquistion
    - When load on frontend is high, the compute node is told to wait before next retry
- Rationale
  - The kickstart file is everything -- without it, a node is just a $2,000 paperweight
  - NACK feature is for supporting large cluster reinstallations

Loader Modifications

- Watchdog
  - If can’t get kickstart file or if there is an error during the installation, reboot
    - This will restart the installation
    - RedHat just halts
- Rationale
  - Again, the kickstart file is everything

Loader Modifications

- Network-based frontend installations
    - In Rocks lingo: a “central” install

- Rationale
    - The “CD dance” during installation is not optimal
    - Needed to support grids of clusters from a central place
    - Huge benefit for development
        - Don’t have to burn CDs just to test code changes

Loader Modifications

- Secure kickstart
  - Added HTTPS support
- Rationale
  - Needed for support of network-based frontend installations (“central” installs)
    - Don’t want the root password for the frontend sent over the network in the clear!
  - Useful for compute nodes that are installed over a public network

Loader Modifications

- Support adding compute node to any ethernet interface
  - The first interface that receives a kickstart file, is anointed ‘eth0’
- Rationale
  - Email reduction
    - We got lots of email from people who plugged their ethernet cable into the “wrong” port
  - Even the Three Stooges can plug in the cables to the compute nodes!

Loader Modifications

- Bug Fixes
  - Added support for multiple CD drives
  - A couple stack overflow problems
- Rationale
  - Without the fixes, the installer halts

In The Next Session

Graph details