



Testing and Debugging rolls

Build a Skeleton Roll

Rocks-A-Palooza I

May 2005

Nadya Williams nadya@sdsc.edu

Prepare for the build

◆ Check out CVS Rocks distribution:

- `cvs -d:pserver:anonymous@cvs.rocksclusters.org:/home/cvs/CVSROOT login`
- `cvs -d:pserver:anonymous@cvs.rocksclusters.org:/home/cvs/CVSROOT checkout -r ROCKS_3_3_0 rocks`

◆ Go to the top-level rolls directory:

- `# cd rocks/src/roll`
- `# ls`

bin/	cac/	etc/	hpc/	patch/
birn/	condor/	gfarm/	intel/	nbc/
birn-oracle1/	CVS/	grid/	java/	ninf/
...				

Create new roll directory

◆ Prerequisites

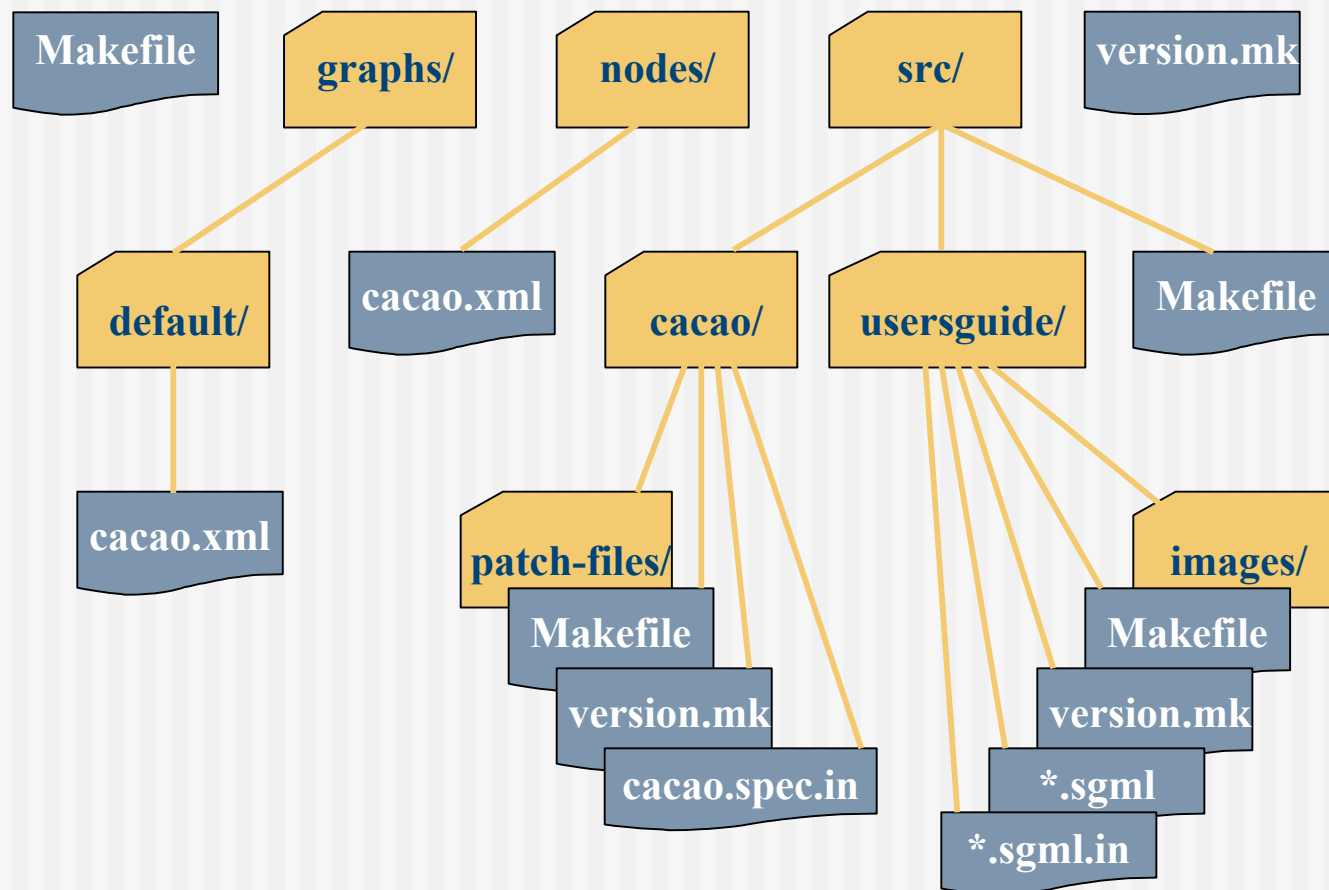
- **name**
- **version** (or defaults to 1)
 - 2
 - 23.4
 - 2a-17beta
- **color** (or random choice from X colors)



◆ Creation

- **# cd rocks/src/roll**
- **# bin/make-roll-dir.py -n cacao -v 15.44 -c chocolate**
- **# ls cacao**
graphs/ Makefile nodes/ src/ version.mk

cacao/ contents



Clean new roll directory

◆ Remove template's id entries

old:

```
# $Id: Makefile,v 1.1 2004/12/01 01:31:55 nadya Exp $
```

new:

```
# $Id: Makefile,v Exp $
```

◆ Remove template's log entries

old:

```
# $Log: Makefile,v $
```

```
# Revision 1.1 2004/12/01 01:31:55 nadya
```

```
# baseline
```

new:

```
# $Log: Makefile,v $
```

What to clean?

- ◆ Makefile
- ◆ graphs/default/cacao.xml
- ◆ nodes/cacao.xml
- ◆ src/Makefile
- ◆ src/cacao/Makefile
- ◆ src/cacao/cacao.spec.in
- ◆ src/usersguide/Makefile
- ◆ src/usersguide/roll-cacao-usersguide.in

File graphs/default/cacao.xml

- ◆ the file describes how all the files in the nodes directory are linked together in the kickstart graph
- ◆ specify edges from known nodes to *cacao* node
- ◆ for graph file syntax see
<http://www.rocksclusters.org/rocks-documentation/reference-guide/3.3.0/kickstart-xml.html#GRAPH-XML>

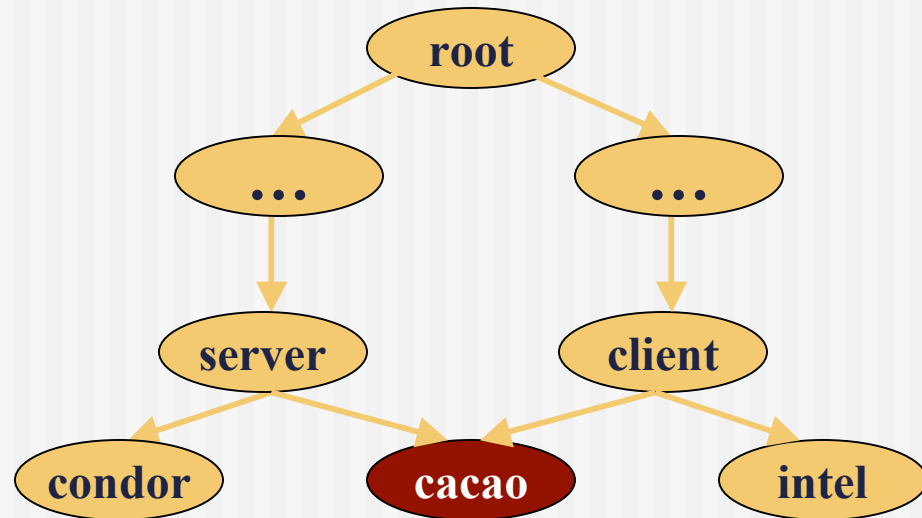
Add edges in graphs/default/cacao.xml

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

```
<graph>  
  <description>  
    cacao Roll  
  </description>  
  
  <changelog>  
    $Log: cacao.xml,v $  
  </changelog>
```

```
  <edge from="server" to="cacao"/>  
  <edge from="client" to="cacao"/>
```

```
</graph>
```





File nodes/cacao.xml

- ◆ the files in nodes/ are used to install packages and to configure their respective services
- ◆ for nodes files syntax see
<http://www.rocksclusters.org/rocks-documentation/reference-guide/3.3.0/kickstart-xml.html>

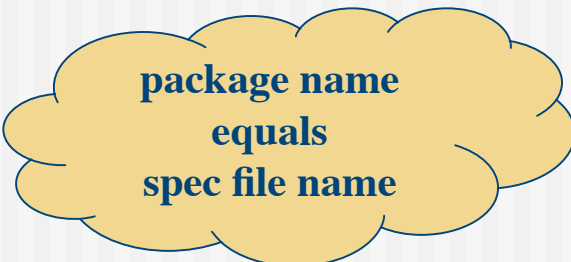
Add packages in nodes/cacao.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE kickstart SYSTEM "@KICKSTART_DTD@">
<kickstart>
  <description>
    Extension to Java beans, for internal consumption only.
  </description>

  <copyright></copyright>
  <changelog>$Log: cacao.xml,v $<changelog>

  <package>cacao</package>
  <package>roll-cacao-usersguide</package>

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



**package name
equals
spec file name**

src/cacao/

◆ # `cp /Venezuela/Chuao/cacao-15.44.tar.gz .`

Makefile cacao-15.44.tar.gz cacao.spec.in version.mk

◆ Makefile

PKGROOT=/opt/cacao

...

install::

mkdir -p \$(ROOT)/\$(PKGROOT)

(cd \$(NAME)-\$(VERSION); make prefix=\$(ROOT)/\$(PKGROOT) install;)

build:

gunzip -c \$(NAME)-\$(VERSION).tar.gz | tar -x

(cd \$(NAME)-\$(VERSION); ./configure --prefix=\$(PKGROOT); make;)

◆ version.mk

NAME = cacao

RELEASE = 0

VERSION = 15.44

Caution!

- ◆ Naming convention is important

cacao-15.44.tar.gz will fail

- ◆ Why?

\$(NAME)-\$(VERSION) combo

NAME = cacao

VERSION = 15.44



cacao-15.44

- ◆ Fix

- NAME = cacao

VERSION = 15.44

cacao_15.44.tar.gz ---> cacao_15.44/

\$(NAME)_\$(VERSION)

- NAME = **Cacao**

VERSION = 15.44

Cacao.spec.in

Adding The 'fortune' Package

```
<?xml version="1.0" standalone="no"?>
<kickstart>
  <description>
  </description>

  <copyright></copyright>
  <changelog>$Log: cacao.xml,v $<changelog>

  <package>cacao</package>

  <package>fortune</package>
  <package>fortune-starwars</package>
  <package>fortune-simpsons-homer</package>

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

Using 'fortune'

```
<?xml version="1.0" standalone="no"?>
<kickstart>
.
.
.
<post>

<file name="/etc/motd" mode="append">
<eval shel="bash">
/usr/bin/fortune
</eval>
</file>

</post>

</kickstart>
```

Linking cacao into the Graph

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

<graph>
  <description>
    The cacao Roll
  </description>

  <order gen="kgen" head="compute">
    <tail>cacao</tail>
  </order>

  <edge from="server" to="cacao"/>
  <edge from="client" to="cacao"/>

</graph>
```

Building

- ◆ **Build rpms**

- `# cd rocks/src/roll/cacao/src/cacao; make rpm`

- ◆ **Build roll**

- `# cd rocks/src/roll/cacao; make roll`

- ◆ **Result**

/home/cacao/rocks/redhat/RPMS/noarch/roll-cacao-kickstart-3.3.0-0.noarch.rpm

/home/cacao/rocks/redhat/RPMS/noarch/roll-cacao-usersguide-3.3.0-0.noarch.rpm

/home/cacao/rocks/redhat/RPMS/i386/cacao-15.44-0.i386.rpm

/home/cacao/rocks/redhat/SRPMS/cacao-15.44-0.src.rpm

/home/cacao/rocks/redhat/SRPMS/roll-cacao-usersguide-3.3.0-0.src.rpm

roll-cacao-3.3.0-0.i386.iso



Deploying a New Roll

Deploying in 3.3.0

◆ Copy the roll into the distro

```
# mount -o loop <rollname>*.iso /mnt/cdrom  
# cd /home/install  
# rocks-dist copyroll
```

Deploying in 3.3.0

◆ Edit “/opt/rocks/etc/rocks-distrc”

⇒ At the bottom, you'll see:

```
<option name="with-roll" value="hpc"/>  
<option name="with-roll" value="kernel"/>  
<option name="with-roll" value="sge"/>  
</rocks-dist>
```

⇒ Add a line:

```
<option name="with-roll" value="hpc"/>  
<option name="with-roll" value="kernel"/>  
<option name="with-roll" value="sge"/>  
<option name="with-roll" value="<roll-name>"/>  
</rocks-dist>
```



Deploying in 3.3.0

- ◆ Rebuild the distro

```
# cd /home/install  
# rocks-dist dist
```

- ◆ The Roll is now available for compute nodes

- ➡ It is also available for frontend “central” installs

Deploying in 4.0.0

- ◆ Similar to 3.3.0, but simpler

```
# mount -o loop <rollname>*.iso /mnt/cdrom  
# cd /home/install  
# rocks-dist --install copyroll
```

- ◆ Roll info now stored in the database
- ◆ The '--install' flag inserts a new row in the database



Testing Your Roll

Testing

- ◆ First, check if there any XML syntax errors:

```
# cd /home/install  
# ./sbin/kickstart.cgi --client=compute-0-0 > /tmp/ks.cfg
```

- ◆ You should see no output
 - But, /tmp/ks.cfg should “look like” a kickstart file



Visualize the Graph

- ◆ Point your web browser to

<https://<frontend name>/>



Visualize the Graph

Rocks-142 Cluster

[Cluster Database \(SSL\)](#) | [\(read-only\)](#)

[Cluster Status \(Ganglia\)](#)

[Cluster Top \(Process Viewer\)](#)

[SGE Job Queue](#)

[News \(RSS\)](#)

[Proc filesystem](#)

[Cluster Distribution](#)

[Kickstart Graph](#)

landscape [sm](#) | [med](#) | [lg](#)

portrait [sm](#) | [med](#) | [lg](#)

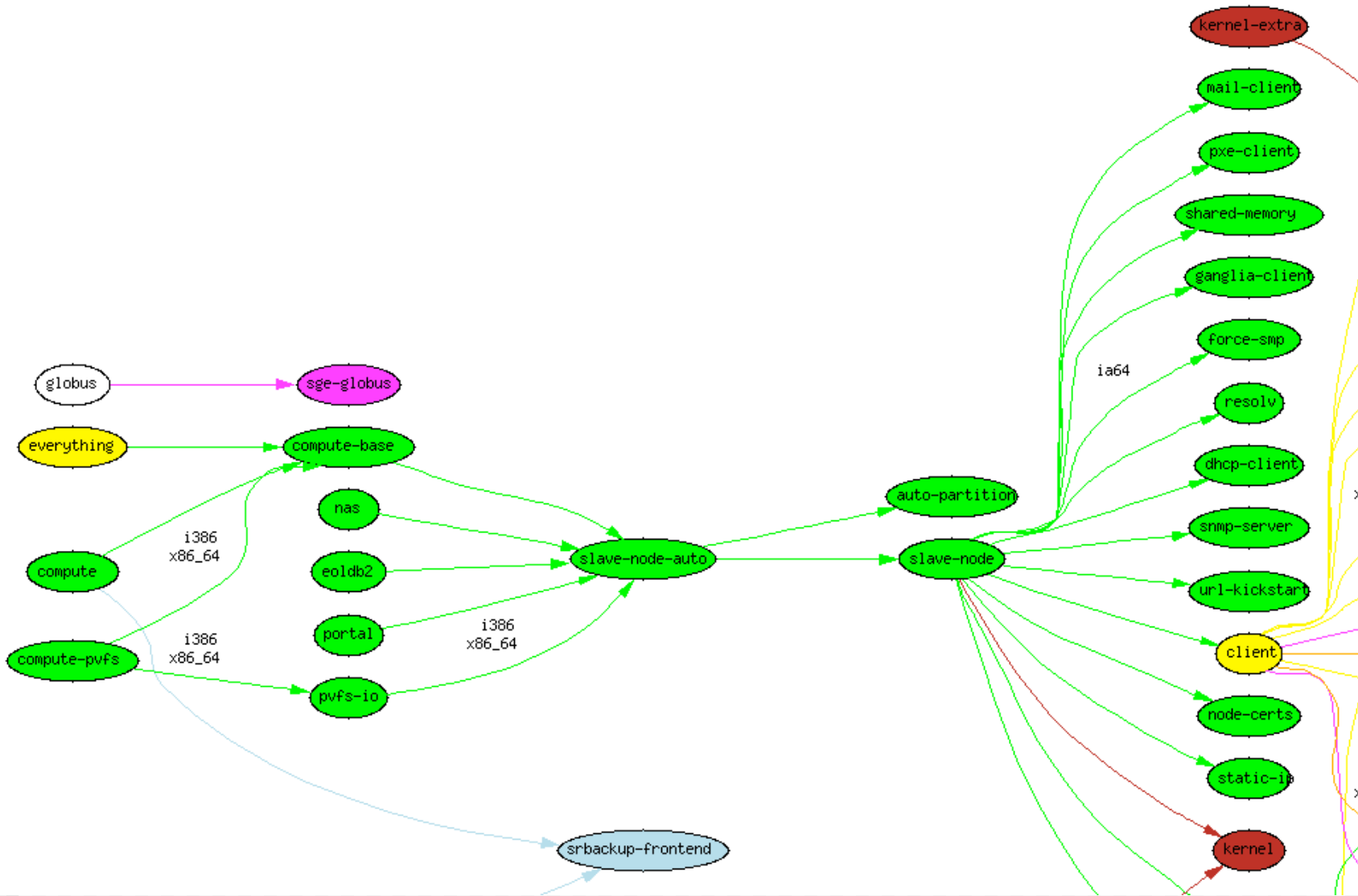
[Roll Call](#)

[Rocks Users Guide](#) | [Reference Guide](#)

Click on one of
these links

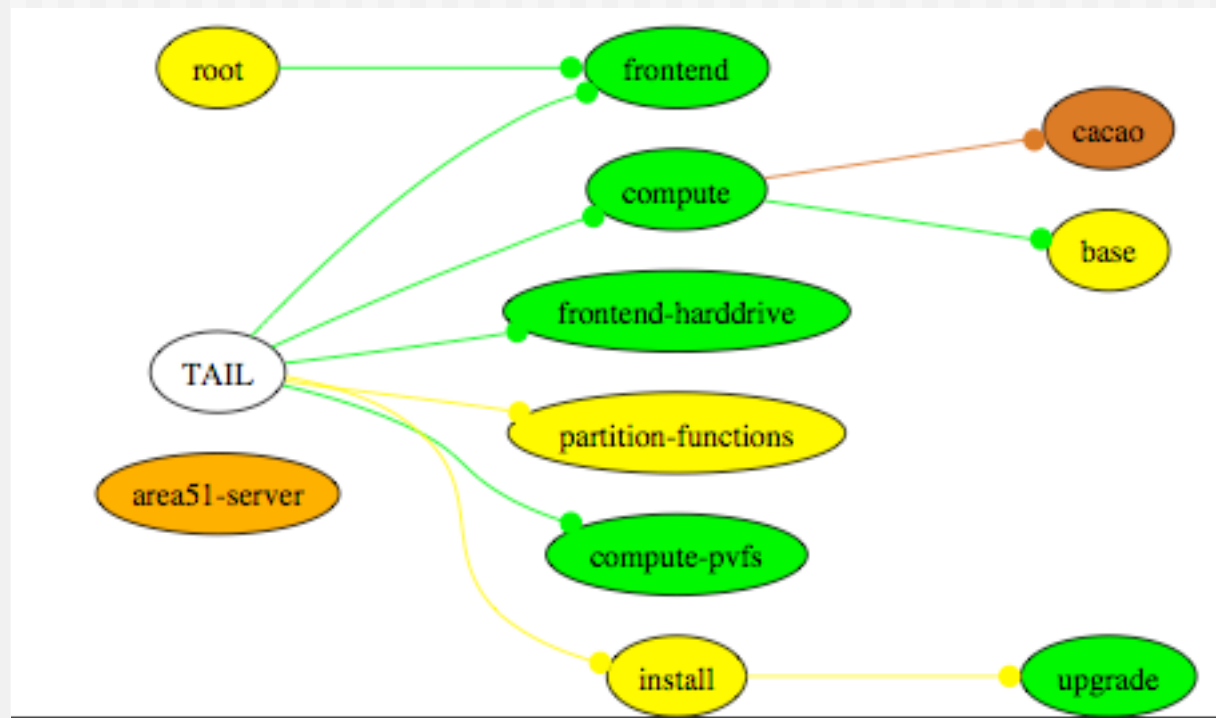


Visualize the Graph



Visualize the Graph

```
# rocks-dist --graph-draw-order --graph-draw-landscape \  
--graph-draw-format pdf graph \  
> /var/www/html/graphs/graph-order.pdf
```



Compute Node Testing

- ◆ To test a compute node, reinstall one

```
# shoot-node compute-0-0
```

- ◆ Connect to the console via “ekv”

```
# ssh -p 2200 compute-0-0
```



Frontend Testing

- ◆ Burn a CD from the ISO image and perform a CD-based install, or
- ◆ Perform a “central” install

Frontend Central Install

- ◆ Prep the “central” server (this is the node that you just installed your roll on)

- ➔ Make sure ‘http’ and ‘https’ ports are open
 - Edit /etc/sysconfig/iptables and uncomment line:

```
# Uncomment the line below to activate web access to the cluster.  
#-A INPUT -m state --state NEW -p tcp --dport www -j ACCEPT
```

- Note: In 4.0.0, you’ll need to enable both http and https
- If you make a change to iptables, restart the service

```
# service iptables restart
```

- ➔ Open access for central install:

```
# insert-access --all
```

Frontend Central Install

- ◆ Boot the frontend with Rocks Base CD
 - ➔ For Rocks 4.0.0, boot with the Kernel Roll
- ◆ At boot prompt, type:
`boot: frontend central=<FQDN of central server>`

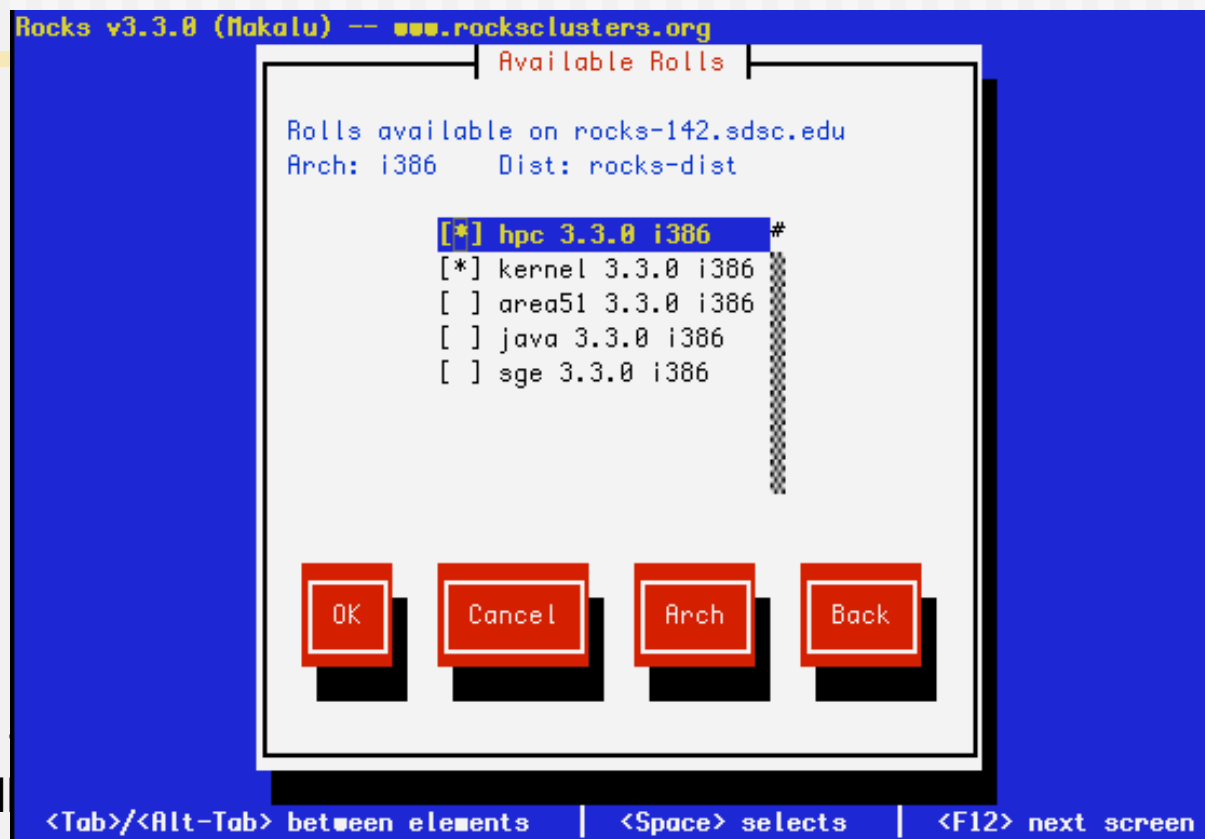
Frontend Central Install

- ◆ Trick: Redirect the console access to the network
 - ➔ That is, we'll apply 'ekv' to a central-based frontend installation:
`boot: frontend central=<FQDN of central server> ekv`
 - ➔ Then, from the central server, connect to installing frontend by:
`# ssh -p 2200 <FQDN of installing frontend>`
 - ➔ Need to be on central server because the public key for root is copied to the installing frontend
 - Ensures only the person with the passphrase for root's private key on the central server can connect to the installing machine



Frontend Central Install

- ◆ You'll be able to select your Roll when you see the following screen:



- ◆ After instal

ntend