Platform Open Cluster Stack (OCS) User Guide

Version 4.1.1-2.0

October 25 2006

Platform Computing

Contents

[Top]

What is Platform OCS?

Building a Linux® cluster is a challenging and time-consuming task. There are many tools in the community and on the Internet for building, configuring, and managing Linux clusters. However, these tools typically assume a familiarity with Beowulf clusters and the concepts of Linux clusters.

Platform Open Cluster Stack (OCS) is a pre-integrated, vendor certified, software stack that enables the consistent delivery of scale-out application clusters. Platform OCS enables a new class of users by simplifying Linux® cluster application, deployment and management. Backed by global 24x7 enterprise support, Platform OCS is a modular and hybrid stack that transparently integrates open source and commercial software into a single consistent cluster operating environment.

This product includes software developed by the Rocks Cluster Group at the San Diego Supercomputer Center at the University of California, San Diego and its contributors. For more information, visit http://www.rocksclusters.org.

Platform OCS is fully supported by Platform Computing Corporation and requires a Red Hat® based operating system such as Red Hat® Enterprise Linux or CentOS Enterprise Linux.

Where to get Platform OCS?

Platform OCS 4.1.1 is released as two different editions: Enterprise and Standard edition. Before installing, make sure you have the following documentation for your particular edition, and have reviewed them before starting your installation:

- Platform OCS 4.1.1 README

- Platform OCS 4.1.1 Installation and Troubleshooting Notes

If you plan on installing other third-party rolls, obtain the CD or DVD containing those rolls.

You can download Platform OCS Standard Edition from the Platform web site at http://my.platform.com/products/platform-ocs.

Contact Platform Computing to purchase Platform OCS Enterprise Edition.

[Top]

Pre-installation

The following steps summarize the Platform OCS pre-installation process:

1. Check the hardware configuration
2. Check the network configuration

Check the hardware configuration

Before Platform OCS is installed, a set of minimal hardware requirements must be satisfied. A typical Platform OCS cluster uses a Beowulf-type cluster setup consisting of the following types of hosts:

Frontend node

The frontend node (or head node) is responsible for the following:

- Administration, managing, and monitoring the cluster
- Installation of compute nodes
- User login, compilation, and submission of jobs to the cluster
- Acts as a firewall to shield the cluster from external hosts and networks
- Acts as a server for many important services: DHCP, NFS, DNS, NTP, HTTP, etc.

Minimal hardware requirements for a frontend are as follows:

- 512 MB of physical memory (RAM)
- 17 GB of free disk space
- Two Ethernet interfaces: one connected to the public network (eth0), and one to the private network (eth1)

Compute nodes

One or more compute nodes are responsible for the following:

- Performing calculations
- Running batch/parallel jobs

Minimal hardware requirements for compute nodes are:

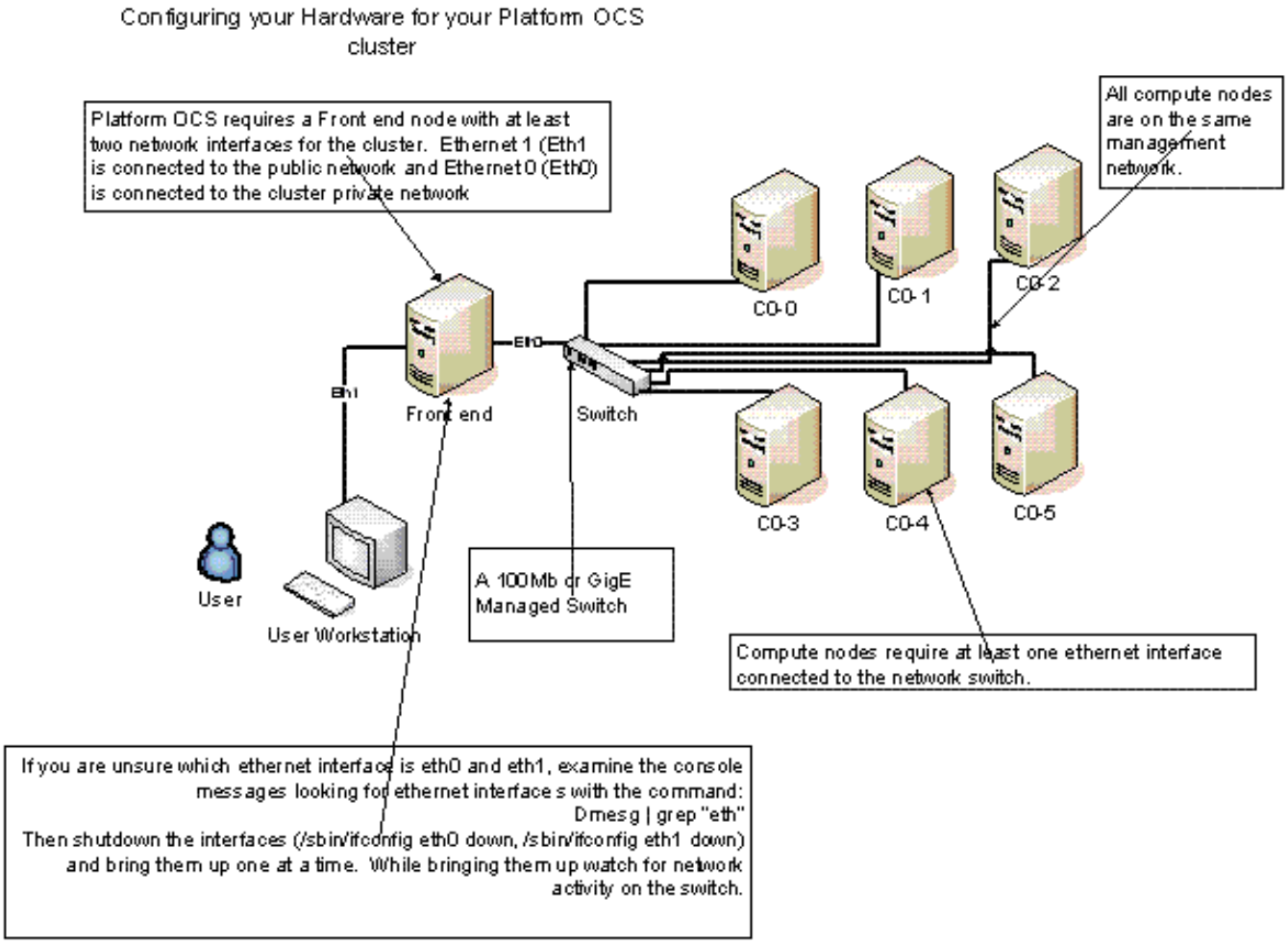
- 512 MB of physical memory
- 17 GB of free disks pace
- One Ethernet interface connected to the private network (eth0)

Optional hardware for compute nodes:

- Additional Ethernet interfaces for connecting to other networks
- Additional Interconnects for High-Performance message passing. Examples: Myrinet and Infiniband

Cluster setup

The following is a diagram that illustrates the cluster setup:



Check the network configuration

In the figure above, the frontend node connects to both a private network through the Ethernet interface mapped to eth0, and to the public network through the Ethernet interface mapped to eth1. The public network refers to the main network in your company or organization. A network switch connects the frontend and compute nodes together to form a completely private network. Other cluster configurations are possible such as exposing all of the compute nodes to the public network by connecting them directly to the public network and not hidden behind the frontend node; however, this type of configuration is not supported at install time.

The private network connecting the frontend and compute nodes is typically a Gigabit or 100Mb Ethernet network. In this simple setup, the private network serves three purposes:

- Cluster administration
- Cluster monitoring
- Message passing

However, it is common practice to perform message passing over a much faster network using a high-speed interconnect such as Myrinet or Infiniband. A fast interconnect provides benefits such as higher throughput and lower latency. For more information about a particular interconnect, please contact the appropriate interconnect vendor.

Testing network configuration

To ensure a successful Platform OCS installation, the Ethernet switches need to be configured properly. There are some installation issues caused by specific switch configurations.

1. If spanning tree is enabled on the switch it dramatically slows down PXE installation because each port in the switch is trying to determine where it fits in the Spanning Tree to avoid loops in the network. Caution should be used when changing the spanning tree configuration options on your switch. A Platform OCS cluster with a single network switch will not need spanning tree configured because there is no possibility of loops in the Ethernet network. However, if multiple switches are required in the cluster then spanning tree is needed to ensure that no loops are created in the Ethernet network topology. Platform recommends disabling spanning tree.
2. Check if PortFast is disabled on the switch. Different switch manufacturers may use different names. It is the forwarding scheme the switch uses. For best installation performance the switch should begin forwarding the packets as it is receiving them. This will speed the PXE booting process. Platform recommends enabling PortFast if it is supported by the switch.
3. Check if Multi-casting is disabled on the switch. Certain switches may need to be configured to allow multi-cast traffic on the private network. Certain tools in Platform OCS such as Ganglia (Cluster Monitoring Tool) require multi-casting enabled to collect information correctly. The switch(es) should be configured for multi-cast traffic for proper Ganglia data collection.
4. Run diagnostics on the switch to ensure the switch is connected properly, and there are no bad ports or cables in the configuration.

Network information

Information about your network is required during installation. Collect the following items from your company or organization's IT department:

- Frontend host information:
  - Hostname (Fully Qualified Domain Name)
  - Static IP address
  - Subnet mask value
  - Gateway address
  - DNS server addresses
- Private Network information:
  - Private IP Address
  - Subnet Mask

[Top]

Frontend node installation

The following steps summarize the installation of Platform OCS on your frontend node:

1. Start the Platform OCS installer
2. Configure your frontend
3. Partition your frontend
4. Test the frontend node

Start the Platform OCS installer

Perform the following steps to start the Platform OCS installer:

1. Insert the Platform OCS DVD into your frontend

After your hardware is setup and connected, you are ready to start installing your frontend.

2. Power up the frontend node with the Platform OCS 4.1.1 DVD. If the DVD does not boot, you must configure the frontend's BIOS to boot from the DVD drive.
3. You will see a splash screen, accompanied by a boot prompt. Type **`frontend`** and press **Enter**. You need to be quick because the installer will start automatically if you do not type anything in the boot prompt within 10 seconds. If you miss typing `frontend`, the installer assumes you are installing a compute node, and not a frontend. Simply power down the frontend and start again.

   After the splash screen, the installer loads the kernel and initial ramdisk. You can abort the loading process by pressing **Ctrl-C** when you see "`Loading vmlinuz...`" or "`Loading initrd.img...`". This returns you to the boot prompt.

4. When you see the **Available Rolls** dialog, you are ready configure your frontend, as specified in Configure your frontend.

Optional steps for booting:

The Platform OCS installer can be booted with optional parameters. Some common boot parameters include:

Boot parameters	Description
dd	This option prompts the user to enter a Driver Disk. When you have hardware that is not supported by the Linux kernel used by the Platform OCS installer, use a Driver Disk to load the kernel drivers for your hardware. Consult your hardware vendor for the Driver Disk.
Other boot parameters	Other boot parameters can be used to alter the boot process. Examples: • Use the mem=XXXM parameter to specify the amount of physical memory to use for the installation (where XXX is the amount in MB). • Use the noacpi parameter to disable ACPI For a full list of boot parameters, refer to the Red Hat Enterprise Linux documentation.

User input

Subsequent installer screens require you to input some values. The following is a list of general tips for navigating between screen elements:

- To navigate between fields, use **Tab** and **Alt-Tab**.
- To push a button, use **Space**.
- Do not press **F12** to advance to the next screen. If you do so, the installer will run incorrectly.
- Most screens have an **Ok** or a **Yes** button to confirm an action or accept input values. Screens also have a **Back** button to go back to a previous screen. You can use this **Back** button to go back and modify a value you typed in.

If you encounter an issue during the installation, you can look for more information in the following locations:

- The first virtual console, `tty1` (**Alt-F1**) is the main console for the installation Installer screens are displayed on this console, including any error messages.
- The second virtual console, `tty2` (**Alt-F2**) is a command-line prompt allowing you to access the OS.
- The third virtual console, `tty3` (**Alt-F3**) displays all of the messages generated by the Platform OCS installer. To view the entire log, you can switch to the command-line, and open `/tmp/anaconda.log`.
- The fourth virtual console `tty4`, (**Alt-F4**) screen displays all of the messages generated by the Kernel. This screen may contain helpful messages to diagnose hardware problems, such as kernel driver issues. To view the entire log, you can switch to the command-line, and run "dmesg".

Configure your frontend

At the **Available Rolls** dialog, select the rolls to install on your frontend, and add your cluster information.

About rolls

A roll groups together packages and configuration scripts that are used to install a specific component for a Platform OCS cluster. For example, a roll can install a batch job scheduler, a driver for an interconnect, or a cluster monitoring package. The DVD contains all of the Rolls you need to install a frontend. There are two types of Rolls:

- Required: These Rolls must be installed for Platform OCS to work. These Rolls are pre-selected and cannot be de-selected by the user.
- Optional: These Rolls install optional components. Users must manually select these Rolls to install them.

Selecting the rolls to install

Complete the following steps to select the rolls to install on your frontend node:

1. In the **Available Rolls** dialog, select all of the rolls you want to install on your frontend. We recommend selecting the Lava Roll (batch job scheduler). The table below is a summary of what rolls are included on the DVD, grouped by category. Choose what you require for your new cluster. Note that your DVD may contain other rolls depending on what Platform OCS edition you have (either Enterprise or Standard edition).

Category	Rolls
Required Platform OCS rolls	Base, HPC, Kernel, OS, Platform
Batch job scheduling systems	Lava, LSF HPC
Interconnects	Cisco™ Topspin®, Myrinet
Cluster monitoring systems	Clumon, Ganglia, Ntop
Parallel file systems	PVFS2
Vendor Customizations	Dell, Intel, HP

2. Press **OK** when you have finished selecting the rolls you wish to install.
3. Select **Yes** to install more rolls on the frontend or **No** if you are finished adding Rolls.

   The installer displays the list of rolls selected for installation from the DVD, and prompts you to enter additional CD or DVD to install more rolls. In most cases, the DVD contains all of the rolls you need for installation. However, if you have more rolls to install, select **Yes**. Otherwise, select **No**.

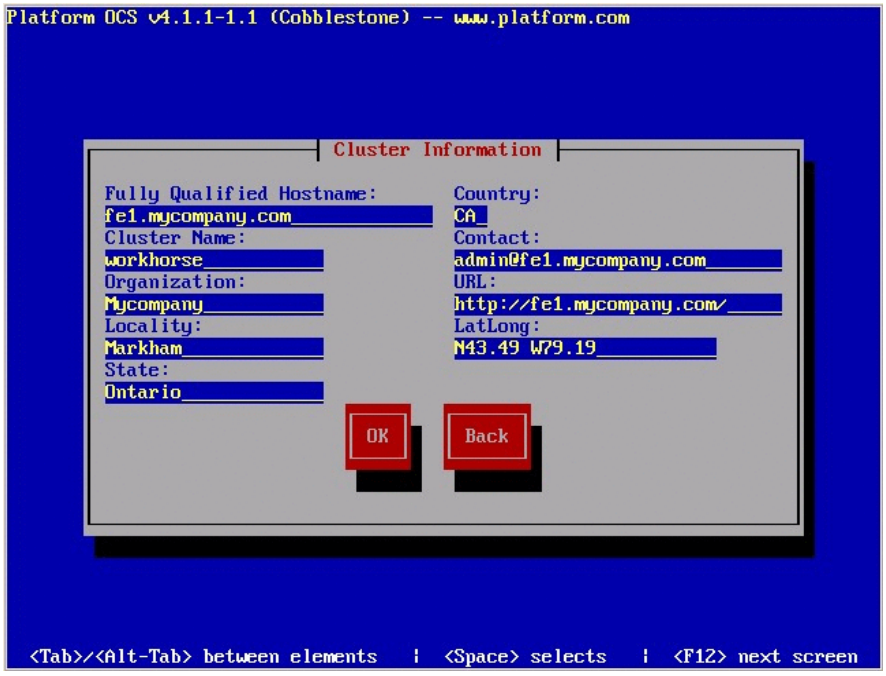
   Links to additional rolls can be found on Platform web site at: http://my.platform.com/products/platform-ocs.

4. Insert the CDs or DVDs containing the additional Rolls:

   Skip this step if you selected **No** in the previous step. Otherwise, perform the following:

   a. Select **Ok** when prompted to insert a CD/DVD.
   b. A roll selection screen similar to the one for the boot DVD is displayed if the CD/DVD you inserted has multiple rolls. A roll selection screen is not shown if a CD/DVD contains one roll or has only required rolls. If so, the installer automatically selects all of the rolls.
   c. The installer will continue to prompt you for more CD/DVDs. Select **No** when you have added all the rolls.
5. In the **Cluster Information** dialog, specify the details of your Platform OCS cluster.

   Enter a Fully Qualified Domain Name (FQDN) for the hostname. The domain name should match your company or organization's domain name.



6. When you see the **Disk Partitioning Setup** dialog, you are ready to partition the hard disk in your frontend, as described in Partition your frontend.

Partition your frontend

Partition the hard disk in your frontend. You need to decide whether to auto-partition your hard disk or manually partition your hard disk.

Auto-partitioning quickly partitions the first disk on your frontend using a default Platform OCS partition scheme. You can select an alternate disk to partition. Auto-partition uses the following partition scheme:

Partition	Mountpoint	Filesystem tyoe	Minimum size	Default size
Root	/	ext3	6 GB	10 GB
Swap	None	swap	1 GB	4 GB
Export	/state/ partition1	ext3	10 GB	Rest of disk

Manual partitioning requires you to manually set up the partition scheme. This includes setting the correct mount-points and specifying appropriate partition sizes.

We recommend Auto-partitioning for most users, You should only select Manual partitioning (Disk Druid) if you want more control over how the disk is partitioned.

At the **Disk Partition Setup** dialog, choose to auto-partition or manually partition your disk:

- To auto-partition your hard disk, follow the steps in Auto-partition your hard disk.
- To manually partition your hard disk, follow the steps in Manually partition your hard disk.

Auto-partition your hard disk

Auto-partition your hard disk using the following steps:

1. At the **Disk Partitioning Setup** dialog, select **Autopartition**.
2. Select the disk to partition and specify whether you want to preserve existing partitions.

   The installer supports three options for preserving partitions on the disk in which Platform OCS is installed.

   - Remove all Linux partitions.

     This preserves any non-Linux partitions, such as Windows partitions (e.g. FAT, FAT32, and NTFS partitions), and any data on those partitions.

   - Remove all partitions.

     This wipes out all partitions on the disk. All data on those partitions will be lost. Note: On Dell systems the Dell utility partition will be preserved.

   - Keep all partitions.

     This preserves all existing partitions, including the data on those partitions. Partitions for Platform OCS are added in the available free space.

     If you choose the option to preserve non-Linux partitions, or all partitions, the amount of free space on the disk must satisfy the minimum required disk space to install a Platform OCS frontend. Refer to the Install checklist for the minimum requirements.

     If there isn't enough space left on the disk, the Platform OCS installer will display an error message to indicate it "Could not allocate requested partitions". The installer will not let you proceed. You have to select **Ok** to reboot the machine.

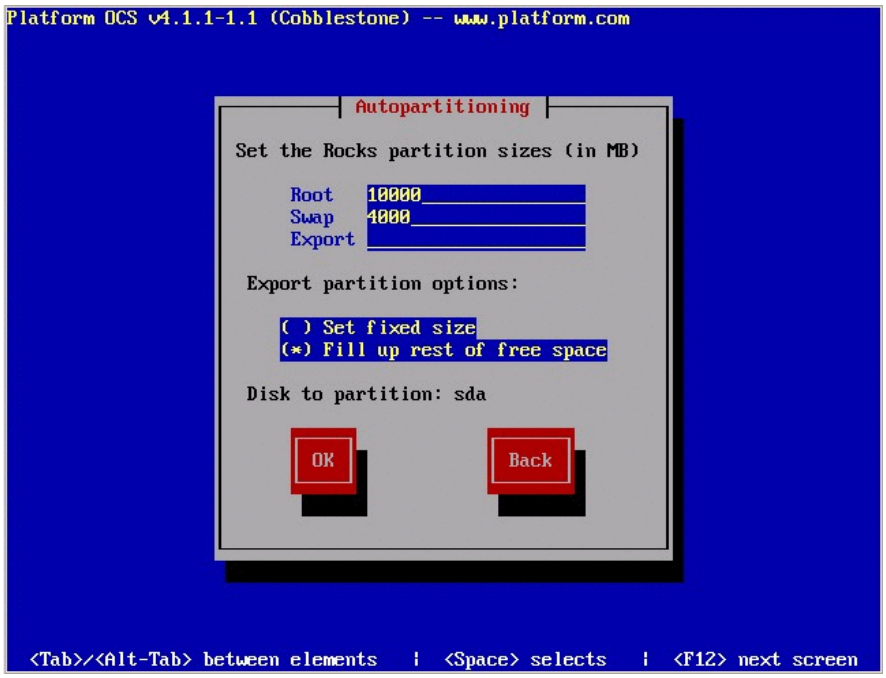
     You can only select one disk to use for the installation. You have the option of selecting any disk currently attached to the machine, including any externally attached disks. If your machine has SCSI or SATA disks, the first disk is named "sda". If your machine has IDE disks, the first disk is named "hda". If you want to partition more than one disk, you have to select Manual Partitioning. You can select **Back** to return to the **Disk Partitioning Setup** dialog and select **Manual Partitioning**, and proceed to Manually partition your hard disk

     Select **Ok** to proceed to the next screen.

3. Specify the sizes for the default Platform OCS partitions.

   The default partition scheme creates a root, swap, and export partition. These partitions are required for Platform OCS to function correctly. The root partition is where the Linux OS is installed, and the export partition is used to store the Platform OCS distribution and the Roll files that it uses.

   You must set the partition sizes in Megabytes (MB). You have the option of setting the export partition size to a fixed size, or make it grow to fill the remaining space on the disk.



Select the **Back** to return to the **Prepare Disk** dialog. Select **Ok** to proceed to the next dialog.

4. Review the automatically created partition layout
    - If you select **No** to review the partitions, you will advance to the **Boot Loader Configuration** dialog. Proceed to Manually partition your hard disk, but skip the first step.
    - If you select **Yes**, you are taken to the **Disk Druid dialog** to verify the partition scheme, and make changes if necessary. Proceed to Manually partition your hard disk.

Manually partition your hard disk

In the **Disk Druid** dialog, verify the partitioning scheme on your hard disk. If you did not choose to auto-partition your hard disk, you need to manually configure the partition scheme in this dialog.

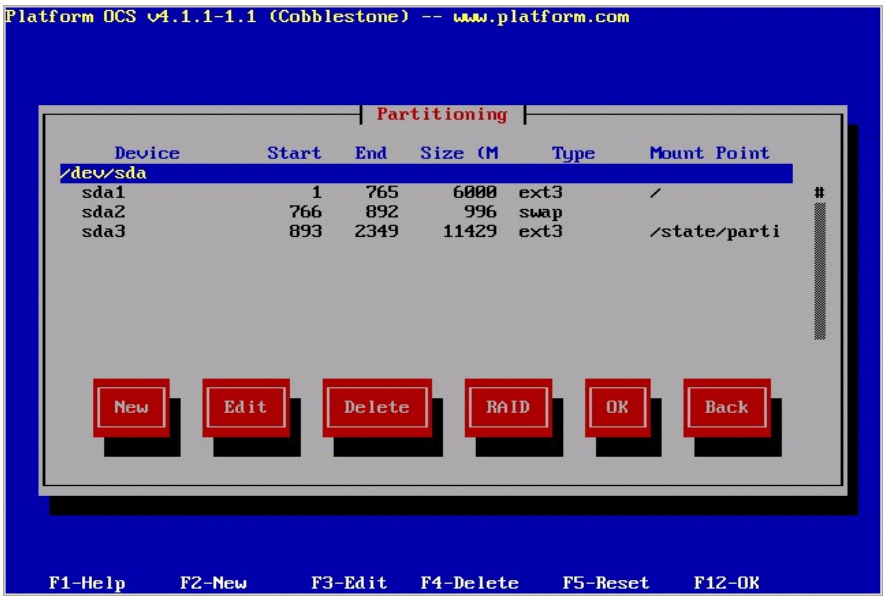
1. Update the partitioning layout with Disk Druid.

    There are two possible paths that can bring you to this screen:

    - You chose Auto-partitioning, and elected to review the partitioning scheme created
    - You chose Manual partitioning

    Disk Druid allows you to create, delete or modify partitions. If you are Auto-partitioning, you can augment the default scheme by creating new partitions. If you are Manually partitioning, you must create the minimum set of partitions required by Platform OCS. This includes the root, swap and export partitions. When you are satisfied with the partition layout, select **Ok**.

    Do not select **RAID** as Platform OCS does not support Software RAID partitioning.



2. Select the default partition to boot for the GRUB boot loader

   The Platform OCS installer automatically adds boot entries to the GRUB boot menu for any operating systems it finds in any partitions that are preserved on the disk. This only occurs if you chose to preserve partitions. Only entries for non-Linux operating systems are added. If you like to add entries to the GRUB menu for Linux operating systems, you must add them manually after the frontend is installed.

   To change the default partition to boot, select the partition and press **F2**. You can also change the label for a partition by selecting it and pressing **Edit**.

Completing the Installation

1. In the **Network Configuration for eth0** and **eth1** dialogs, specify the IP address for the Private (eth0) and Public (eth1) Ethernet interfaces

   A Platform OCS frontend requires two Ethernet interfaces to work correctly. The next two screens ask the user to enter the IP address and Netmask for the private and public interfaces.

   For the private interface, only class-based networks are supported. Classless Inter-Domain Routing (CIDR) is not supported (e.g. subnetting or supernetting). The following is a list of valid Netmask values and the number of hosts each Netmask value supports. Choose the Netmask value that is appropriate for your cluster size.

Class	Netmask value	Number of hosts in the network
A	255.0.0.0	16777214
B	255.255.0.0	65534
C	255.255.255.0	254

   For the public interface, you need to contact your IT department to obtain a static IP address for the frontend, and the corresponding Netmask value. You cannot configure the frontend to use an IP address obtained via DHCP.

2. In the **Miscellaneous Network Settings** dialog, specify your gateway and DNS IP addresses.

   You may need to contact your IT department to obtain these addresses for your network.

3. In the **Time Configuration** dialog, select your time zone from the list of servers and specify your network time server. If your node uses UTC time, select **System clock uses UTC**.

4. In the **Root Password** dialog, select a root password that you will remember.

The installer will format the disk, copy the rolls from the DVD (and any other CD/DVDs you inserted) onto the disk, and install the packages.

After package installation completes, the boot loader is installed and the post-installation is executed. The machine then reboots. You have completed your installation, and are ready to test your frontend node as described in Test the frontend node.

Test the frontend node

Before installing the compute nodes, perform the following tests to verify your frontend is operational. Log in to your frontend as root with the password you used during the installation and perform the following steps:

1. Check for hardware issues

   In some cases, you might have hardware that is not detected by the running kernel, or you have a kernel driver that fails to load. Look through the following logs to identify any hardware issues:

   a. Check the kernel logs for any hardware driver issues or other errors:

      ```
      # dmesg
      ```

   b. Check the system logs for any startup issues or other errors:

      ```
      # less /var/log/messages
      ```

2. Check that the ethernet network is working:
   a. Check that both eth0 and eth1 interfaces are up:

      ```
      # ifconfig
      ```

   b. Verify the routing table is correct.

      ```
      # route
      ```

      When verifying the routing table, pay careful attention to the following:

      - Traffic for the private network is routed over eth0, while traffic for the public network is routed over eth1.
      - The default route will go through the gateway server you specified during installation.
      - Multicast packets will be routed over eth0 (using 224.0.0.0 network)
      - External hosts can be reached with the ping command
3. Check that the High Performance Interconnect is working

   If you installed an interconnect, you should verify the driver for the interconnect hardware was loaded correctly. In addition, the interconnect vendor may provide diagnostic tools to determine if the interconnect is working. We suggest you refer to the documentation for your particular interconnect.

4. Check the required services

   The frontend runs many services that are essential for cluster administration and installing compute nodes. You need to ensure all of the services listed below are running:

Service	How to check

Web Server	`service httpd status`
DHCP	`service dhcpd status`
DNS	`service named status`
Xinetd	`service xinetd status`
MySQL database	`service mysqld status`
NFS	`service nfs status`
AutoFS	`service autofs status`

5. Check the Platform OCS infrastructure

   Run some basic Platform OCS commands, seen below, to verify the infrastructure is working. The commands should execute successfully.

   a. Login as root and start `insert-ethers`, select compute node, then press F11 to exit.

      # **insert-ethers**

      Important: If you run "insert-ethers", you might see a message that says "Rocks Distribution is not ready. Please wait for rocks-dist to complete". This is normal when you log into a frontend for the first time. A startup script runs rocks-dist in the background during the first boot-up. You have to wait for "rocks-dist" to finish running before you can run "insert-ethers".

      ---

   b. Test rebuilding the Platform OCS distribution

      # **cd /home/install ; rocks-dist dist**

6. Check the added rolls

   Verify that all of the rolls you selected during the frontend installation are added to the frontend:

   # **rollops -l**

   You can use the "rollops" command to install other rolls from the DVD. Simply insert the DVD, and run the following command. This command will display a menu from which you can select the roll you want to install:

   # **rollops -a**

7. Start up X Windows

   Run the following command to start X Windows:

   # **startx**

   This command will automatically probe for your video card, configure the settings for it, and start up X. It may be necessary to run **`system-config-display`** to configure the display correctly. You can configure Platform OCS to automatically start X every time you log in by changing the runlevel on the `initdefault` line from 3 to 5 in the `/etc/inittab` file.

8. Check the Platform OCS Cluster home page

   Verify that you can access the cluster home page. The page will load automatically when you start the browser. This Homepage gives you access to all of the Cluster Monitoring tools, and Platform OCS documentation for all of the installed rolls.

   Follow the link near the bottom of the Homepage to register your Platform OCS cluster.

When all the above tests pass, you are ready to proceed with compute node installation. If you experience any issues or errors, contact Platform Support at support@platform.com.

[Top]

Compute node and appliance installation

Different types of nodes can be installed in a Platform OCS cluster. These different node types are referred to as appliances. The most common type is a compute node. The other appliances are listed in the table below. The set of available appliances will depend on what rolls you install. You can view the list of available appliances by running the insert-ethers command.

Appliance Type	Installed by	Description
Compute	Base Roll	Creates a standard compute node. Other Appliance types are based on this basic Compute appliance type.
LSF HPC Master	LSF HPC Roll	Creates an LSF HPC Master Candidate Host for fail-over of the master host in an LSF HPC cluster.
Pvfs2-meta-server	PVFS2 Roll	Creates a PVFS2 Meta Server th3 at maintains the distributed file system index for PVFS2, and a PVFS2 Data Server.

Ethernet Switches	Base Roll	Use this if you have a managed Ethernet switch. It is used to assign an IP address to a managed switch. This is done so that DHCP requests from a managed switch are not confused with DHCP requests from a compute node.

Note: Platform OCS provides an optional method to install compute nodes that involves pre-loading host information into the Platform OCS database to speed up the compute node installation process. This also allows system administrators to pre-configure the cluster naming and IP scheme making it independent of the order in which nodes are installed. This method requires a list of MAC addresses. To take advantage of this feature, you must obtain a list of MAC addresses for your compute nodes before installing the compute nodes.

The following steps summarize the installation of Platform OCS on your compute nodes and appliances:

1. Prepare your compute node
2. Install compute nodes
3. Install other appliance types
4. Test compute nodes and appliances
5. Test the cluster installation

There are two methods for installing compute nodes and other appliance types: using the `insert-ethers` tool, or using the `add-hosts` tool. Choose the method that is appropriate for your cluster.

About insert-ethers

Insert-ethers is a tool you run on your frontend to capture the DHCP requests broadcasted by the compute nodes. For each DHCP request, insert-ethers generates a hostname and IP address for the node and adds the new information to the Platform OCS database.

The system is then updated to reflect the addition of the new host. Various system configuration files are updated, and DHCP and DNS services are restarted. Once DHCP is updated, a compute node can obtain an IP address, allowing it to network boot, and start the install process. Insert-ethers should be used if you are deploying a small to medium sized cluster (less than 128 nodes). Insert-ethers uses a node naming convention based on the assumption that your nodes are assembled in racks. The convention is:

```
<appliance type>-<rack>-<rank>
```

where:

- `<appliance type>` is the short-form name of the node's appliance type
- `<rack>` is the number of the rack in which the node is located
  - Value starts from 0.
- `<rank>` is the location within the rack where the node is located
  - Value increases from the bottom of the rack to the top of the rack
  - Value starts from 0.

For example:

- `compute-0-0`: This is a compute node that is located in the bottom-most node in the first rack.
- `lsfhpc-5-5`: This is a LSF HPC master candidate host that is located in the sixth row (from the bottom) in the sixth rack.

The `insert-ethers` command assigns IP addresses to nodes starting from the top-most IP address for your subnet, and iterates through the address space in descending order.

For example, given a frontend address of 10.1.1.1, and a netmask of 255.0.0.0, the first node is assigned 10.255.255.254, the second node is assigned 10.255.255.253, and so on.

About add-hosts

Add-hosts is a tool that pre-populates the Platform OCS database with host information. The tool enables the user to define their own hostnames and IP addresses for the compute nodes using an XML configuration file. This alleviates the need to run insert-ethers to capture DHCP requests, and auto-assign hostnames and IP addresses.

After the information is loaded into the database, the system is updated to reflect the addition of the new hosts, in the same way that `insert-ethers` updates the system. `add-hosts` should be used if you are deploying a large cluster of greater than 128 nodes. `add-hosts` requires a list of MAC addresses for your compute nodes. If you are purchasing new hardware for the cluster the hardware vendor can supply a list of MAC addresses for all nodes.

Prepare your compute node

Before installing your compute nodes, consider customizing them to suit your requirements. The most common customizations are:

- Changing the default partition layout
- Adding additional RPM packages
- Adding additional post-installation configuration scripts

To customize your compute nodes, you need to update the Platform OCS distribution. Customizations are specified using XML files. Every change to an XML file requires a rebuild of the Platform OCS distribution.

The Platform OCS distribution is located in `/home/install/rocks-dist`. To rebuild it login as root and, run:

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

Important: Always rebuild the distribution in the `/home/install` directory. Rebuilding the distribution in other directories may result in corruption of the permissions in the `/home/install` directory.

The XML files for compute node customization are located in `/home/install/site-profiles/4.1.1/nodes`. The XML files can be generated manually or generated using automated tools included with Platform OCS. Details are described in the next section.

1. Changing the default partition layou**t**

   You can change the default partition sizes, or create your own partition layout to override the default Platform OCS partition layout. The default partition layout for compute nodes is the same as the layout for the frontend. Only the first disk is partitioned, other disks are left as is.

Partition	Mountpoint	Filesystem Type	Minimum size	Default size
Root	/	Ext3	6 GB	10 GB
Swap	None	Swap	1 GB	4 GB
Export	/state/ partition1	Ext3	10 GB	Rest of disk

   a. Changing the default partition sizes

      If you're satisfied with the default layout, but want to change the root and swap partition sizes, use the custom-partition tool:

      ```
      # custom-partition -r <root partition size in MB> -s
      <swap partition size in MB> -b
      ```

For example, to change the root partition size to 20 GB, and swap partition size to 2 GB, run the following command:

```
# custom-partition -r 20000 -s 2000 -b
```

The "custom-partition" tool creates the `/export/home/install/site-profiles/4.1.1/nodes/extend-a uto-partition.xml` file and rebuilds the Platform OCS distribution.

For more information about the custom-partition tool, refer to the manpage or the Readme for Platform OCS Rolls.

b. Changing the default partition layout

To setup more complex partitioning, you need to manually create a `replace-auto-partition.xml` file that will replace the default layout and rebuild the Platform OCS distribution.

Run the following commands:

```
# cd /home/install/site-profiles/4.1.1/nodes
# cp skeleton.xml replace-auto-partition.xml
```

Open `replace-auto-partition.xml` with a text editor and:

i. Delete the <package> and <post> sections
ii. For each partition you want to define, create a line with the <part> tag in between the <main> and </main> tags
iii. Between the <part> and </part> tags, specify the parameters for your partition. The parameters used are the same as those used for the RedHat Kickstart "part" directive.
iv. For more information on the different partition parameters, please refer to the Advanced Partitioning section of this guide.

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

For example:

Suppose you want to create a partition layout on the first SCSI disk consisting of a 15 GB root partition, 2 GB swap partition, 5 GB `/var` partition, and a `/data` partition that takes up the rest of the disk. Here is what the XML file will look like:

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

<description>
</description>

<changelog>
</changelog>

<main>
<!-- Put your partitioning directives here -->
  <part> / --size 15000 --ondisk sda </part>
  <part> swap --size 2000 --ondisk sda </part>
  <part> /var --size 5000 --ondisk sda </part>
  <part> /data --size 1 --grow --ondisk sda </part>
</main>

</kickstart>
```

2. Adding additional RPM packages

   The rocks-compute tool can be used to update the Platform OCS distribution with the user's own RPM packages. The tool allows you to add, list, or remove packages. The tool creates the `/export/home/install/site-profiles/4.1.1/nodes/extend-comp ute.xml` file and rebuilds the Platform OCS distribution. You can add as many packages as needed.

   - To add a custom package to the Platform OCS distribution and rebuild the distribution, run the following:

     # **rocks-compute -a -p** <*path to the RPM package*> **-b**

     Important: the rocks-compute tool does not check for RPM package dependencies for a given package. Ensure that you also add run the command above for any package dependencies.

   - To list all of the packages you added:

     # **rocks-compute -l p**

     The above command will list a unique ID for each package. This ID is used to remove a package from the distribution.

   - To remove a package from the Platform OCS distribution and rebuild the distribution, run:

     # **rocks-compute -d -p** <*package ID*> **-b**

   Example: Adding your own RPM package

   # **rocks-compute -a -p /myshare/package-1.0.0.x86_64.rpm -b**

   Example: Adding an RPM from the OS roll

   The OS roll contains RPMs for the Linux operating system. There may be some RPMs in the OS roll that you want to install but didn't get installed on the compute node. The steps are:

   a. Look for the RPM you want to install. Let's install the "ncompress" package:

      # **find /home/install/ftp.rocksclusters.org/pub/rocks/rocks-4.1.1/rocks-dist/rolls/os/4.1.1/x86_64/RedHat/RPMS/ -name 'ncompress*'**

   b. Add the `ncompress` package to the Platform OCS distribution

      # **rocks-compute -a -p /home/install/ftp.rocksclusters.org/pub/rocks/rocks-4.1.1/rocks-dist/rolls/os/4.1.1/x86_64/RedHat/RPMS/ncompress-4.2.4-40.x86_64.rpm -b**

3. Adding additional post-installation configuration scripts

   You may want to add your own post-installation scripts to configure a compute node. Some examples include turning on/off services, creating or updating configuration files, creating init scripts, etc.. The scripts are executed during the post-installation after all RPM packages have been installed. Create the script in a text

editor, and save it to a file. The script must be a bash shell script.

The `rocks-compute` tool can be used to update the Platform OCS distribution with the user's post-installation scripts. The tool allows you to add, list, or post-installation scripts. The tool creates the `/export/home/install/site-profiles/4.1.1/nodes/extend-c ompute.xml` file and rebuilds the Platform OCS distribution. You can add as many scripts as needed.

- To add a post-installation script and rebuild the Platform OCS distribution:

  ```
  # rocks-compute -a -s <path to script> -b
  ```

- To list the post-installation script(s) you added:

  ```
  # rocks-compute -l s
  ```

The above command will list a unique ID for each script. This ID is used to remove a script from the distribution.

- To delete a post-installation script, and rebuilding the Platform OCS distribution:

  ```
  # rocks-compute -d -s <script ID> -b
  ```

  Example: Adding a post-install script

  Suppose you have a Bash script that appends a library path to the `/etc/ld.so.conf` file. You can create a script that looks as follows.:

  ```
  #!/bin/bash
  echo "Appending /mypath/lib to /etc/ld.so.conf" >> /root/compute.log
  echo "/mypath/lib" >> /etc/ld.so.conf

  echo "Running ldconfig" >> /root/compute.log
  /sbin/ldconfig >> /root/compute.log 2>&1
  ```

  Let's assume the script is saved in `/home/user/myscript.sh`. You can run add the script by running:

  ```
  # rocks-compute -a -s /home/user/myscript.sh -b
  ```

  For more information about the rocks-compute tool, refer to the manpage or the Readme for Platform OCS Rolls.

Install compute nodes

Install compute nodes using either `insert-ethers` or `add-hosts` as follows:

- Installing compute nodes using insert-ethers
- Installing compute nodes using add-hosts

Installing compute nodes using insert-ethers

1. Log in to the Platform OCS frontend as root, and run `insert-ethers`.
2. If you have a managed ethernet switch that sends out DHCP requests, select **Ethernet Switches**. If you didn't, proceed to the next step.

   Choosing **Ethernet Switches** will assign an IP address to the switch. You may need to wait several minutes for the switch to broadcast a DHCP request. When done, press **F9** to quit.

3. If you are installing a small cluster and you are not worried about assigning hostnames and IP addresses in the same order as the physical host layout, just run `insert-ethers`.

   If you do care about order, you need to tell the `insert-ethers` command which rack you are installing by specifying the rack number on the command-line. Let's assume you want to start with the first rack:

   ```
   # insert-ethers --cabinet=0
   ```

   The nodes will be named `compute-0-0`, `compute-0-1`, `compute-0-2`, and so on.

4. Choose **Compute** from the list of appliances.
5. Once `insert-ethers` is waiting for the compute node, you can PXE boot the node by either physically rebooting the node from the console or remotely logging into the console using Vendor IPMI or management tools.

   To make sure that your compute nodes are assigned hostnames and IP addresses in the correct order, you will need to PXE boot each machine, one at a time, in order of their physical location in the current rack you are installing. In other words, power up the bottom-most node in the rack, then work your way up, one node at a time.

6. If the node is successfully detected, installation will begin and you should see the MAC address and compute node name on the `insert-ethers` screen.

   An asterisk `(*)` indicates that a kickstart file was requested by the compute node and installation should proceed normally. If there is no `(*)`, Platform OCS will not install properly on the node and you should see an error on the compute node.

   Once a node has a `(*)`, you can PXE boot the next node in the rack. Repeat the process for the rest of the nodes in the rack.

   If you see a `(503)` status, it means that the frontend is too busy to serve a Kickstart file to a node. In this case, try PXE booting the compute node again. If you see a "(500)" status, then an error occurred when generating the Kickstart file for the node. In this case, verify whether the Kickstart file can be generated locally on the frontend.

7. You can monitor the installation of a compute node by either switching to the console of the compute node with a kvm switch or using management tools supplied by the hardware vendor. If you do not have a kvm switch or you have not configured the hardware management utilities you can still monitor the installation progress of the compute node by creating a secure shell connection to the compute node.

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

8. You should see the install progress on the compute node.
9. Once installation is complete, the node will automatically reboot and join the cluster.
10. Once you have finished installing all of the compute nodes in rack `0', exit `insert-ethers' and run it again incrementing the cabinet number, for example:

    ```
    # insert-ethers --cabinet=1
    ```

    Return to Step 4 and repeat the installation process for the rack.

Installing compute nodes using add-hosts

For small clusters insert-ethers is the quickest and easiest way to install Platform OCS. However, for larger clusters of 128 nodes and beyond, the `add-hosts` tool provides better configuration management. A large cluster requires planning out the layout of the network, switches, racks, and nodes in the cluster. The `add-hosts` tool is an easy

way to plan out the cluster layout, when a list of all of the MAC addresses is provided by the hardware vendor.

The steps are as follows:

1. Obtain a list of the MAC addresses for all nodes in the cluster. Save the addresses in a text file. For example, /opt/rocks/etc/mac.txt. The MAC addresses must be listed in the order in which you plan to add the hosts. In other words, the first MAC address corresponds to the first node in the first rack, the second MAC to the second node in the first rack, and so on.
2. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` to define the names, IP addresses, and appliances that you will be installing.
3. For brevity, we will show you a sample add-hostsrc file and MAC address file, and ask that you refer to the Advanced Administration section of this guide for more information on setting up the `add-hostsrc` file.

   Suppose that you are installing 5 compute node, located in the same rack, in a class B network (i.e. netmask is 255.255.0.0). Assume that you want to assign IP address starting from 10.1.1.5. Your MAC address file and add-hostsrc file will contain:

   MAC Address file:

   ```
   00:11:22:33:44:55        # first compute node
   00:11:22:33:44:56        # second compute node
   00:11:22:33:45:57        # etc......
   00:11:22:33:45:58
   00:11:22:33:45:59
   ```

   Add-hostsrc file:

   ```
   <?xml version="1.0" standalone="yes"?>

   <add-hosts>
     <mac_addr_file value = "/opt/rocks/etc/mac.txt" />
     <num_hosts_per_rack value = "10" />
     <order_by_rack value = "yes" />
     <netmask value = "255.255.0.0" />

     <subnet>
       <host_prefix value = "compute" />
       <baseip value = "10.1.1.5" />
       <num_hosts_in_subnet value = "5" />
       <appliance value = "compute" />
     </subnet>

   </add-hosts>
   ```

4. Run `add-hosts` to populate the database based on the information in the XML file above

   ```
   # add-hosts
   ```

   The following information is added to the Platform OCS database:

Hostname	IP Address	MAC Address
compute-0-0	10.1.1.5	00:11:22:33:44:55
compute-0-1	10.1.1.6	00:11:22:33:44:56

compute-0-2	10.1.1.7	00:11:22:33:44:57
compute-0-3	10.1.1.8	00:11:22:33:44:58
compute-0-4	10.1.1.9	00:11:22:33:44:59

5. PXE boot your compute nodes. Note that the order in which your nodes are PXE booted is not important since the node information is already in the database. You can PXE boot several hosts at the same time.

Install other appliance types

In addition to compute nodes, you can install other appliance types:

- Install an LSF HPC master candidate host
- Install a PVFS2 meta server

Install an LSF HPC master candidate host

If you installed the LSF HPC roll, you can install LSF HPC master nodes to fail-over the LSF HPC master host to another host. This increases cluster uptime and availability. We recommend installing one or more LSF HPC master nodes if you are setting up a large cluster.

Install an LSF HPC master candidate host using the following steps:

1. Log into the frontend as root
2. Run `insert-ethers` and select the LSF HPC Master appliance type.
3. Install one or more of the `LSF HPC Master` nodes using PXE boot.
4. Exit `insert-ethers` by pressing F9 to update the `lsf.cluster.lsfhpc` file.
5. Create an NFS shared path on another NFS server, and make sure that this NFS path can be mounted on the new LSF HPC master node.
6. On the frontend, run the following:

```
# cd /home/install/upgrades/lsfhpc
# config-lsf-master
```

7. Answer the dialog questions when prompted by the script.

Install a PVFS2 meta server

If you installed the PVFS2 roll, you can install this appliance type. The PVFS2 appliance installs a server that acts as both a PVFS2 Meta Server and Data Server. It will create a sample PVFS2 filesystem that is mounted under `/mnt/pvfs2`.

Install a PVFS2 meta server using the following steps:

1. Log into the frontend as root
2. Run `insert-ethers` and select the `Pvfs2-meta-server` appliance type.
3. Install the PVFS2 meta server using PXE boot
4. Repeat the process till all the nodes to be used as Data Servers are installed.
5. Exit `insert-ethers` by pressing F9.
6. Follow the instructions in the PVFS2 Roll section under Production Cluster Configuration to complete the configuration.

Test compute nodes and appliances

You can test the compute nodes and appliances as follows:

1. Check if you can log into the compute node without a password:

   ```
   # ssh <compute node name>
   ```

2. Check DNS by resolving the frontend's hostname:

   ```
   # host <frontend's local name>
   ```

3. Check if `/home/install` is auto-mounted:

   ```
   # ls /home/install/
   ```

4. Check if 411 can update all of the files on the compute node:

   ```
   # 411get --all
   ```

5. If you installed an LSF HPC master candidate host, perform the following tests:
   a. Make sure the license is installed.
   b. Run the compute node tests
   c. Run the `lsid` command to check that the cluster is up
   d. Run `lsadmin ckconfig` and `badmin ckconfig`. There should be no errors.
6. If you installed a PVFS2 meta server, check that the `/mnt/pvfs2` path is mounted. Test that other compute nodes can also mount the `/mnt/pvfs2` path.

Test the cluster installation

Before proceeding further, make sure you have completed the post-install tests for the frontend and compute nodes. When done, run the following tests to ensure that your cluster is functioning properly.

1. Run Cluster-fork to verify that all nodes can be connected

   ```
   # cluster-fork hostname
   ```

2. Test 411 to verify that 411 broadcasts can be sent out to all nodes

   ```
   # make -C /var/411 force
   ```

3. Check Ganglia (if installed). Point your browser to **`http://localhost/ganglia`** and verify that all nodes appear on the webpage.
4. Check Clumon (if installed). Point your browser to **`http://localhost/clumon`** and verify that all nodes appear on the webpage.
5. Check Lava cluster (if installed) to see if all nodes appear in the cluster

   ```
   # lsid
   # lsload
   # bhosts
   ```

6. Check LSF HPC cluster (if installed) to see if all nodes appear in the cluster

   ```
   # lsid
   # lsload
   # bhosts
   ```

[Top]

Basic Administration

The following topics describe basic tasks when administrating your Platform OCS cluster:

Online documentation

Online documentation is provided online by the frontend node. Start a browser on the frontend. It will default to the Cluster page, which contains links to the following guides:

- Roll User Guide
- Platform OCS User Guide
- Reference Guide (an SDSC document)

Roll-specific documentation is available from the Installed Rolls link, by following the Guide or Readme links beside the roll of interest.

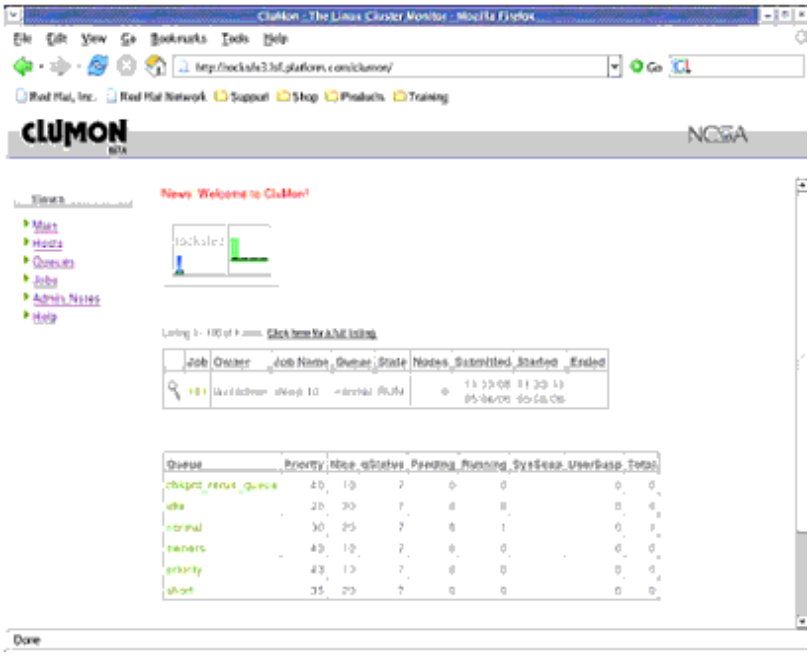
Clumon

Clumon is a cluster job-monitoring tool that allows the administrator to see: the states of jobs, view job queues, load information, resource usage and process information and if a node's scheduler daemons are up or down.

Clumon represents the system load by colours on a bar for each compute node. Icons representing nodes will have indicators denoting the load of the node, red indicating high or heavy load, and various levels of blue to indicate a lighter load. If a node is experiencing problems or is down, the node will become a black and red crossbones icon. If you move your mouse over a node icon, a popup note will appear and provides summary information about the node.

To view Clumon information, go to the main cluster webpage, click on "Cluster Status (CluMon)" link, or point your browser to **`http://localhost/clumon`** (on the frontend).

In a screen with running jobs, you can examine each job's state by clicking on the job number, or by examining the queues:

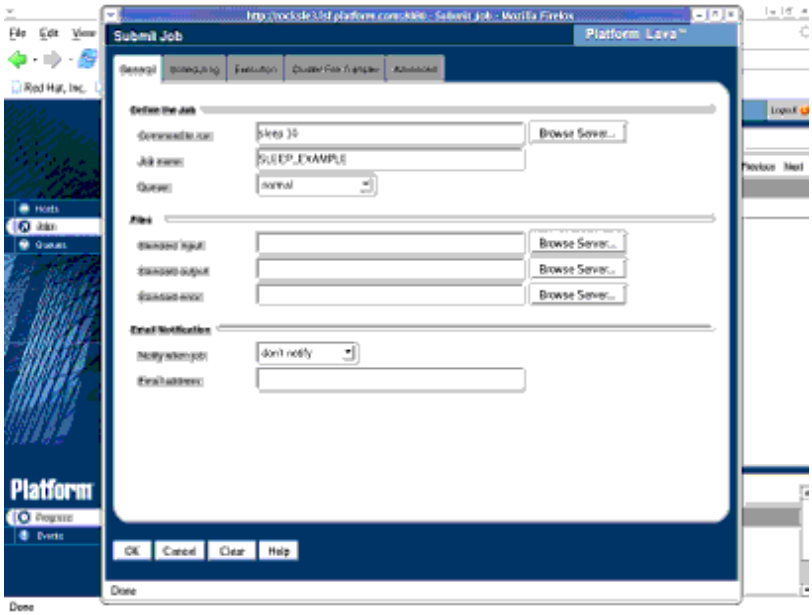


Platform Lava GUI

The Lava web GUI is a frontend to the Lava batch scheduling system. Users can submit jobs and perform actions such as suspending, resuming or killing jobs.

To submit or modify jobs go to the Lava GUI web interface. Go to the main cluster webpage, click on "Lava GUI". You will need to log into the interface. User `root` is not permitted to login. Log in to the interface using an existing user account or the `lavaadmin` account.

The following is a Lava GUI dialog window for submitting a job:



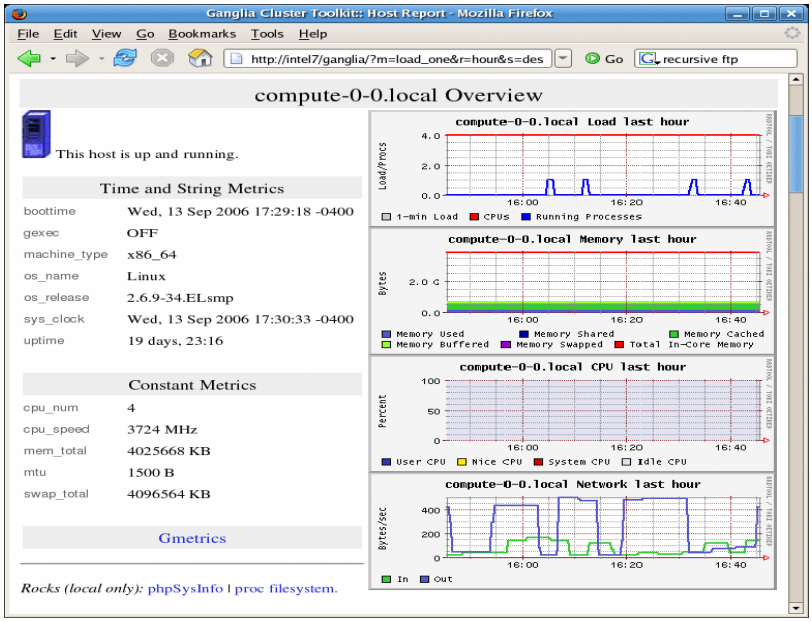
Ganglia

Ganglia is a cluster statistics collector which monitors node availability, displays system load, network usage, and other resource information over a period of time. Data is collected for each metric and is stored on the frontend. The data is stored for up to one year.

Ganglia displays detailed information regarding the usage of each node and provide the administrator a guide as to

the day-to-day functions of the cluster.
To view Ganglia information, go to the main cluster webpage, click on "Cluster Status (Ganglia)" link.

The following is a Ganglia display showing the overview of a cluster:



Ntop

Ntop is a network traffic analyzer designed to show the administrator the different protocol traffic passing through the frontend.. Ntop can also show network traffic patterns to better diagnose network problems and network utilization issues.

By default, Ntop is configured for both public and private traffic with one interface always listening. You can switch which network interface ntop should listen on by clicking the Admin menu option and selecting "Switch NIC". From there a new screen will appear and you can then select which network interface to listen on.

Ntop provides several plug-ins that can be enabled or disabled for further analysis of the traffic. See the plugins page within Ntop for more details.

To view the Ntop page, go to the main cluster webpage, click on "Ntop Cluster Monitoring (SSL)" link.

The following is an Ntop display showing active TCP and UDP sessions connected to a frontend on a private network:

SSH

By default, the OpenSSH daemon is configured to enable X11 forwarding. This can sometimes slow down connecting to nodes. You can disable forwarding by using the `-x` option when connecting to a node to skip X11 forwarding.

This can also be disabled permanently by editing the `/etc/ssh/ssh_config` file and changing the line `ForwardX11 Yes` and setting this to `No`.

An SSH connection from one node to another may be slow in setting up. This is usually because of a name resolution failure, and subsequent timeout. This can occur if the frontend was installed with an invalid DNS server.

Note: this will also slow MPI jobs.

Adding, removing, or upgrading rolls

Platform OCS provides a tool that allows the user to do roll maintenance on their frontend.

Adding a roll

Using the rollops tool, you can add a roll to the frontend. To do this, you need a CD/DVD roll or you can download an ISO image.

1. Insert the CD/DVD roll into the drive or use the -i option to rollops
2. Run either of the following:
    - For a regular CD/DVD roll:

      ```
      # rollops -a
      ```

    - For an ISO image roll:

      ```
      # rollops -a -i isoimage
      ```

   Example output:

   ```
   rollops: Copying Roll: ntop
   Copying roll from media (directory "/tmp/tmprcwC0V") into
   mirror
   ```

```
Copying "ntop" (4.1.1,x86_64) roll...
7645 blocks
chmod a+rx /home/install/ftp.rocksclusters.org
Installing Roll: ntop, please wait...
<Roll installation output>
rollops: The 'ntop' roll has been successfully installed!
```

Note: If the CD/DVD roll or the ISO image is a meta-roll (a roll that contains many rolls in one), you will see a list of rolls to install.

```
rollops: Autodetecting CD-ROM/DVD roll...

Rolls found

1) clumon
2) extras
3) ganglia
4) lsfhpc
5) modules
6) myrinet
7) ntop
8) pvfs2
9) ts_ib

q) Quit

To install a roll, type the number or type "q" to quit>
```

Upgrading a roll

1. Insert the CD/DVD roll into the drive or use the `rollops -i` option.
2. Run either of the following:
   - For a regular CD/DVD roll:

     ```
     # rollops -u
     ```

   - For an ISO roll:

     ```
     # rollops -u -i isoimage
     ```

Example output:

```
rollops: Copying Roll: dell
Copying roll from media (directory "/tmp/tmprcwC0V") into
mirror
Copying "dell" (4.1.1,x86_64) roll...
7645 blocks
chmod a+rx /home/install/ftp.rocksclusters.org
Installing Roll: dell, please wait...
<Roll installation output>
rollops: The 'dell' roll has been successfully upgraded!
```

Note: If the CD/DVD roll or the ISO image is a meta-roll (a roll that contains many rolls in one), you will see a list of rolls to perform an upgrade.
You can upgrade to rolls with the same version or with a newer version but cannot rollback to an older roll.

```
rollops: Autodetecting CD-ROM/DVD roll...

Rolls found

1) myrinet
2) dell
3) intel_mpirt
4) ts_ib

q) Quit

To upgrade a roll, type the number or type "q" to quit>
```

Removing a roll

To remove the roll from the frontend, run the following command:

```
# rollops -e <roll_name>
```

Example output:

```
rollops: Removing Roll: 'ntop', please wait...
<Roll removal output>
rollops: The 'ntop' roll has been removed successfully!
```

Disabling a roll

To disable a roll from being installed on a compute node run the following command:

```
# rollops -p no -r <roll_name>
```

Example output:

```
rollops: Setting permissions for the 'ntop' roll. Please
wait...
rollops: Completed updating permissions for the 'ntop' roll.
```

Adding or removing users

To add a user or delete a user, you must be logged into the frontend as root. After a user is added or removed, 411 automatically updates the user information on all of the nodes in the cluster.

Adding a user

- To specify the password on the same command-line:

  ```
  # adduser -p <password> <user_name>
  ```

- To specify the password using the `passwd` command:

  ```
  # adduser <user_name>
  # passwd <user_name>
  # make -C /var/411
  ```

Removing a user

To remove a user, run the following command:

```
# userdel <user_name>
```

Firewall/iptables

The frontend is installed with firewalling software (iptables). It is configured with some basic forwarding rules. From a network security standpoint the frontend and nodes are not secure. Evaluate the security risks at your site and create appropriate firewall rules to secure the cluster.

Warning: The frontend should never be connected to the Internet without first restricting the type of packets allowed by customizing the iptables rules.

By default, services are only visible to the private network. However, you may choose to enable HTTP and HTTPS over the public network. Please note that this will expose your cluster homepage and Platform OCS database to the external network. To open HTTP and HTTPS access, edit the `/etc/sysconfig/iptables` file and uncomment the following lines:

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

Then, restart iptables:

```
# service iptables restart
```

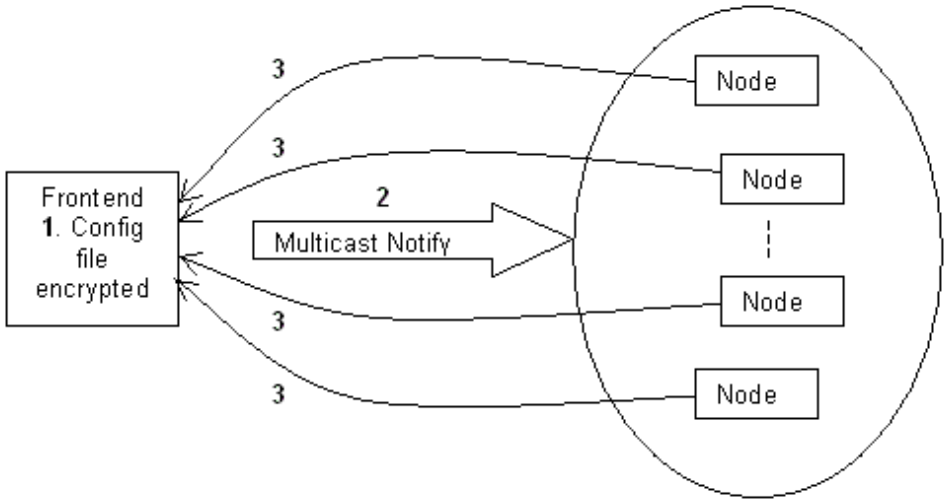
For details on customizing your firewall, see http://www.netfilter.org.

The default routing of Platform OCS is to use eth0 for private and eth1 for public traffic.

Platform OCS services and utilities

411

Platform OCS provides a service called 411. This is very similar to NIS. It is used to synchronize files across a cluster. This is done via multicasting a notification of change from the frontend then having the nodes download the file over an encrypted channel. Users and groups are one example of information passed over 411. Whenever you run `useradd` or `userdel`, 411 will update the user information on all nodes in the cluster.The diagram below depicts the process.



By default the following files are propagated throughout the cluster by 411:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`
- `/etc/services`
- `/etc/rpc`
- `/etc/auto.*` files (for example, `auto.master` and `auto.home`)

If you have made any changes to the files listed above. Running the command `make -C /var/411` will push the updated files to the cluster.

You can also have compute nodes pull any 411 synchronized files by running `411get --all` on the compute node to retrieve all files. To update all compute nodes, run the following command:

```
# cluster-fork 411get --all
```

See the Advanced Administration for how to customize 411.

Rocks-grub

The `rocks-grub` service is a tool that forces an appliance such as a compute node to reinstall if the node is powered off incorrectly, such as a power outage. If the service is turned ON, the node will be reinstalled from scratch. If the service is turned OFF, the node will not be reinstalled on restart.

If you don't want this behavior, you can permanently turn off the service on every node by running: the following:

```
# cluster-fork service rocks-grub stop
# cluster-fork chkconfig rocks-grub off
```

The following appliances have rocks-grub disabled by default: NAS, PVFS2 and LSF Master.

DHCP and tftp

Platform OCS uses DHCP and the TFTP services to handle installation/reinstallation of appliances. The services are automatically configured when running the insert-ethers tool. Insert-ethers configures the DHCP settings for each appliance.

`cluster-fork` provides the ability to execute commands on a cluster wide basis. For example:

```
# cluster-fork 'cat /etc/rocks-release'
compute-0-0:
Rocks release 4.1.1-2.0 (Cobblestone)
lsfhpc-0-0:
Rocks release 4.1.1-2.0 (Cobblestone)
pvfs2-meta-server-0-0:
Rocks release 4.1.1-2.0 (Cobblestone)
compute-0-1:
Rocks release 4.1.1-2.0 (Cobblestone)
lsfhpc-0-1:
Rocks release 4.1.1-2.0 (Cobblestone)
compute-0-2:
Rocks release 4.1.1-2.0 (Cobblestone)
compute-0-3:
Rocks release 4.1.1-2.0 (Cobblestone)
```

Shoot-node

The `shoot-node` command forces a node to reboot and reinstall. You can specify more than one appliance to reinstall on the command line. To use shoot-node, you must start an SSH Agent and pass the name of the shell you are using (usually with the $SHELL variable). For example,

```
# ssh-agent $SHELL
# ssh-add
# shoot-node compute-0-0
[compute-0-0] waiting for machine to go down
[compute-0-0] waiting for machine to go down
[compute-0-0] waiting for machine to come up
[compute-0-0] waiting for machine to come up
[compute-0-0] launching xterm
[compute-0-0] waiting for machine to go down
[compute-0-0] waiting for machine to go down
[compute-0-0] waiting for machine to come up
[compute-0-0] waiting for machine to come up
[compute-0-0] done. (8.34252 minutes)
```

Reinstalling compute nodes

Before you can start reinstalling a compute node, some preparatory may be required. Platform OCS has two reinstallation modes for compute nodes:

- Fresh reinstallation. In this mode, the partitions on the disk(s) that Platform OCS is installed on are re-created, and re-formatted. Choose this method if you made changes to your partitioning layout, either by using the `custom-partition` tool to adjust the root/swap partition sizes, or by making changes to the `replace-auto-partition.xml` file.
- Upgrade. In this mode, only the root partition is re-formatted. Other partitions are preserved (such as `/state/partition1`) and re-mounted. Choose this method if you did not make any changes to your partitioning layout.

Fresh reinstallation:

1. Log into the frontend as root.
2. Remove the `/.rocks-release` file on the compute node you want to reinstall.

   The `/.rocks-release` file tells the Platform OCS installer whether Platform OCS was installed on the host. If it is installed, a compute node upgrade is done instead of a fresh installation.

3. Run the following command to remove the file on the node:

```
# ssh <node_name> rm -fr /.rocks-release
```

4. Remove the partitions for the compute node from the Platform OCS database.

   When the installer decides how to partition a disk for a compute node, it first determines if the database contains the partitioning layout for the node. If the information exists, it will use that layout to partition the disk. If the layout in the database matches the layout currently on the disk, no partitioning changes are made. If there is a mismatch, all of the partitions are recreated using the layout from the database.

   Remove the partitioning information from the database by running the following command:

   ```
   # rocks-partition -list -delete -nodename <node_name>
   ```

5. Make sure you've finished your customizations to the Platform OCS distribution, and rebuild the distribution:

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

Upgrade

1. Log into the frontend as root
2. Make sure the `/.rocks-release` file exists on the node you want to reinstall.

   ```
   # ssh <node_name> ls -l /.rocks-release
   ```

3. If the file does not exist, create it:

   ```
   # ssh <node_name> touch /.rocks-release
   ```

4. Make sure you've finished your customizations to the Platform OCS distribution, and rebuild the distribution.

Reinstallation methods

You are now ready to reinstall your compute nodes. There are three methods to choose from:

1. PXE boot the node

   You can reboot the machine and initiate a PXE boot, or you can hard reboot the machine. A hard reboot involves powering off the machine without performing a proper reboot. After a hard reboot, the node will reinstall automatically.

   Note that you can disable this behavior of reinstalling after a hard reboot on your compute nodes by disabling the **rocks-grub** service. Please see the Platform OCS Services section of this guide.

2. `shoot-node` command

   If you don't have physical access to a node, run the following command to reinstall a node. This command will block until the compute node finishes reinstalling and reboots. The steps to run the command are: as follows

   ```
   # ssh-agent $SHELL
   # ssh-add
   # shoot-node <node_name>
   ```

3. `cluster-kickstart` command

   This is another method for reinstalling a node to which you don't have physical access. Unlike `shoot-node`, this command is non-blocking. When it runs on the compute node, it will instruct the compute node to reboot

itself and reinstall:

```
# ssh <node_name> /boot/kickstart/cluster-kickstart
```

Like a regular installation, you can also monitor the installation of a compute node by creating a secure shell to the compute node:

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

Log files

Platform OCS generates the following logs:

1. Roll installation logs

   Each roll installed may produce a log file in `/root`. For example, the Platform roll produces a log in `/root/platform.log`.

2. System logs

   System logs may be found in the `/var/log` directory. Some important logs include:

   - System log (`/var/log/messages`)
   - Installation log (`/var/log/anaconda.log`)
   - Kernel log (`/var/log/dmesg`)
   - Web server logs (`/var/log/httpd/*_log`)
   - Xorg log (`/var/log/Xorg.0.log`)

3. Tool logs

   The **`custom-partition`** and **`rocks-compute`** commands will produce logs in the directory they were run. The **`rocks-update`** command produces the `/var/log/rocks-update.log`

Cron jobs

Platform OCS provides a cron job to back up the database. The script can be found in `/etc/cron.daily/backup-cluster-db`. A Ganglia cron job also exists and backs up the RRD generated data. This is executed weekly and can be found in `/etc/cron.weekly/ganglia-save-rrds.cron`.

It is assumed the user understands how to modify or add a cronjob. This is out of the scope of this guide.

[Top]

Advanced Administration

The following topics describe basic tasks when administrating your Platform OCS cluster:

- Troubleshooting the installation process
- 411
- IP tables and routing
- Mysql database
- Custom kernel installation

Rocks-update

Platform OCS includes a tool called `rocks-update` that allows you to update your cluster with the latest security updates from the Red Hat ® Network and the CentOS® Network.

Downloading an update

1. Run the following command:

   `# rocks-update -d <package_name>`

2. For Platform OCS Enterprise Edition, if you have not registered with Red Hat Network, you will be prompted to register. When the registration is complete, `rocks-update` will download the requested update.

   NOTE: For Standard Edition, no registration will be performed and `rocks-update` will download the requested update.

   Example output for Enterprise Edition:

   ```
   # rocks-update -d gnupg
   rocks-update: You are not currently registered with the Red
   Hat Network, proceeding to register. Press ctrl+c to
   cancel. Press <Enter> to continue...
   ```

3. Input your Red Hat® account information, including your username, password, and the email address you used when registering with Red Hat®.
4. Click **Next** to continue.
5. Once signed into the Red Hat® Network, click **Next** to continue.

   Do not de-select any of the packages listed for download.

6. Click **Next** when prompted to confirm your settings.

   Your registration information will be sent to Red Hat and a confirmation screen will appear informing you have been successfully registered.

   Once completed, `rocks-update` will then proceed to download your update**.**

```
rocks-update: Checking for updates (this may take some time)...
rocks-update: Determining if any Packages and Dependencies are needed...
rocks-update: Running up2date to download packages...

Fetching Obsoletes list for channel: rhel-x86_64-as-4...

Fetching rpm headers...

Name                   Version    Rel
-------------------------------------------------------------
gnupg                   1.2.6     3         x86_64


Testing package set / solving RPM inter-dependencies...
gnupg-1.2.6-3.x86_64.rpm:  Retrieved.
rocks-update: End of up2date execution...

WARNING: Ensure that the the above up2date output contains no warnings or
errors.
Type "y" to continue or "n" to abort> y
rocks-update: Rebuilding Yum repository...
rocks-update: Synchronizing Rocks distribution (this may take a few minutes)
...
rocks-update: Platform OCS repository for updates is now 4.1.1.1.
```

   You can now install/update your compute node or frontend appliances.

Patching a frontend

Run the following command:

```
# rocks-update -f
```

This will patch the frontend with any updates you have previously downloaded.

For example,

```
# rocks-update -f
rocks-update: Rebuilding Yum repository...
rocks-update: Patching frontend, please wait...
rocks-update: Installed Package: gnupg-1.2.6-3.x86_64.rpm
rocks-update: 1 Update(s) installed successfully on frontend!
```

Patching an appliance

Run the following command:

```
# rocks-update -c <num_update_nodes>
```

The default is to perform updates on 64 nodes at a time. You can change this default by specifying the number of nodes to update concurrently, up to a maximum of 250 nodes.

For example,

```
# rocks-update -c
rocks-update: Patching 64 appliances in group...
rocks-update: Rebuilding Yum repository...
rocks-update: Patching appliance node compute-0-0, please
wait...
rocks-update: Installed Package: gnupg-1.2.6-3.x86_64.rpm
rocks-update: 1 Update(s) installed successfully on
compute-0-0
rocks-update: Successfully patched compute-0-0 (1 of 1)
packages installed!
```

Checking cluster installed version

To get an overview of the repository version and installed versions in the cluster, run `rocks-update -l`. The information displayed is the current repository version on the frontend, the current installed version for the frontend and all the supported appliances.

For example,

```
# rocks-update -l
Current Repository Version: 4.1.1.1
Current Front End Install Version: 4.1.1.1

Appliance      Version
=========      =======
compute-0-0     4.1.1.1
compute-0-1     4.1.1.1
compute-0-2     4.1.1.1
compute-0-3     4.1.1.1
compute-0-4     4.1.1.0
compute-0-5     4.1.1.0
compute-0-6     4.1.1.0
```

Add-hosts

One way to install a compute node is to use the `insert-ethers` command. This command does four things:

- Look for DHCP requests from compute nodes when they PXE boot
- For each request, insert host information for the node into the Platform OCS database. The information includes MAC address, IP address, hostname, and so on.
- Update various system configuration files
- Restart DHCP and DNS services

After this is done, a compute node is able to obtain an IP address using DHCP, allowing a node to PXE boot over the network. This method of capturing DHCP requests and restarting the services for each host has some drawbacks:

- Hosts must be booted in the correct order to enable insert-ethers to assign the correct name and IP address to each node.
- Hosts cannot be installed in batches. This increases deployment time for large clusters having hundreds of hosts.
- Users do not have an easy way to assign custom hostnames and IP addresses.

Platform OCS includes a tool called `add-hosts` to address these drawbacks. It allows you to define the hostnames and IP addresses in an XML file. The tool reads that file to pre-populate the Platform OCS database. With the host

information pre-loaded, compute node installation is a matter of simply powering up your machines.

To use this tool, you will need the following:

- To login as the root user on the frontend
- A list of MAC addresses. This list can be obtained from your hardware vendor.
- To make sure `insert-ethers` is not running when you run the tool

Configuration of `add-hosts` requires two files:

- A text file (`/opt/rocks/etc/mac.txt`) storing all of the MAC addresses, listed in the order in which you will add the hosts.
- An XML file (`/opt/rocks/etc/add-hostsrc`) that defines host information for:
  - One or more individual hosts (`<host>` section)
  - A group or subnet of hosts (`<subnet>` section)

When the `add-hosts` tool runs, it parses the XML configuration file in tandem with the MAC address file. Each host defined in the XML file is mapped to the next occurring MAC address in the MAC address text file. To ensure that each host is mapped to the correct MAC address, the MAC addresses must be listed in the same order as the hosts are defined in the XML configuration file.

Creating the configuration files

The example below will help you get started in guiding you through the process of creating the required configuration files. Suppose that you have threehosts that you wanted to add:

Hostname	MAC Address	IP Address	Appliance Type
compute-0-0	00:11:22:33:44:55	10.2.1.1	Compute
compute-0-1	00:11:22:33:44:56	10.2.1.2	Compute
nfs-server	00:11:22:33:44:57	10.2.1.3	Compute

Before defining an actual XML and MAC address file, let's start with an abstract example. Notice how each host is mapped to the correct MAC address in the file.

XML File	Maps to	MAC Address File
# List individual hosts first		# MAC addresses for individual hosts
nfs-server		00:11:22:33:44:57
# List groups of hosts next		# MAC addresses for groups/subnets of hosts
compute-0-0		00:11:22:33:44:55
compute-0-1		00:11:22:33:44:56

Create the actual XML and MAC address files, and assume that each host is an individual host using `<host>` sections. The meaning of the rest of XML tags will be explained later in this section.

XML File/opt/rocks/etc/add-hostsrc	Maps to	MAC Address File/opt/rocks/etc/mac.txt
<?xml version="1.0" standalone="yes"?>		# MAC addresses for # individual hosts

<add-hosts> <mac_addr_file value = "/opt/rocks/etc/mac.txt" /> > <num_hosts_per_rack value = "10" /> <order_by_rack value = "yes" /> <netmask value = "255.0.0.0" /> <host> <name value = "nfs-server" /> <ip value = "10.1.2.3" /> <appliance value = "compute" /> </host> <host> <name value = "compute-0-0" /> <ip value = "10.1.2.1" /> <appliance value = "compute" /> </host> <host> <name value = "compute-0-1" /> <ip value = "10.1.2.2" /> <appliance value = "compute" /> </host> </add-hosts>	MAC addr #1 MAC addr #2 MAC addr #3	00:11:22:33:44:56 # MAC addresses for # groups/subnets of hosts 00:11:22:33:44:55 00:11:22:33:44:56

The XML file defines the host information that you will add to the Platform OCS database. Note how the host sections are ordered so that each host is mapped to the correct MAC address. For all intents and purposes, this sample XML file will work for a small number of hosts. However, this is not a scalable approach for adding hundreds of hosts to your cluster.

You can logically group `compute-0-0` and `compute-0-1` into one host grouping because the hosts have:

- The same prefixes
- The same appliance types
- Their IP addresses are sequential.

You can express groupings of hosts with similar characteristics by creating a `<subnet>` section. The nfs-server host can be defined on its own since it doesn't have a matching prefix. Using `<subnet>` sections, our example XML file now looks like this:

XML File/opt/rocks/etc/add-hostsrc	Maps to	MAC Address File/opt/rocks/etc/mac.txt
<?xml version="1.0" standalone="yes"?> <add-hosts> <mac_addr_file value = "/opt/rocks/etc/mac.txt" /> > <num_hosts_per_rack value = "10" /> <order_by_rack value = "yes" /> <netmask value = "255.0.0.0" /> <host>		# MAC addresses for # individual hosts 00:11:22:33:44:56 # MAC addresses for # groups/subnets of hosts 00:11:22:33:44:55 00:11:22:33:44:56

<name value = "nfs-server" /> <ip value = "10.1.2.3" /> <appliance value = "compute" /> </host>	MAC addr #1	
<subnet> <host_prefix value = "compute" /> <baseip value = "10.1.2.1" /> <num_hosts_in_subnet value = "2" /> <appliance value = "compute" /> </subnet>		
</add-hosts>	MAC addr #2, then #3	

Add-hosts requires you to define some global parameters:

- `<mac_addr_file>` defines the absolute path to the MAC address file
- `<num_hosts_per_rack>` defines the number of hosts in your rack
- `<order_by_rack>` is a flag to tell add-hosts whether to name hosts by rack and rank value, or to name them using a single integer value.

  For example, If set to "yes", your hosts will be named `compute-0-0`, `compute-0-1`, etc.. If set to "no", your hosts will be named `compute-0`, `compute-1`, and so on.

- `<netmask>` is the netmask for your subnet. Only class-based netmask values are valid. For example, this value is set to 255.255.255.0 if you have a class C network.

The `<host>` section tells the `add-hosts` to add a single host with the given name, IP address, and appliance type.

The `<subnet>` section tells `add-hosts` to:

- Add two compute nodes to the Platform OCS database as specified by `<num_hosts_in_subnet>` and `<appliance>`.
- Use the following naming convention for the name of each host added to the database:

  `<host_prefix>-rack-rank.`

  By default, the rack and rank values start from zero. They're calculated as follows:

  - Rack = floor [(N -1) / <num_hosts_in_subnet>]

    Where N is the Nth host defined in the <subnet> section

  - Rank = (N - 1) modulo <num_hosts_in_subnet>

    Where N is the Nth host defined in the <subnet>

In our example, `<num_hosts_per_rack>` is 10, and `<num_hosts_in_subnet>` is 2. So, Rack is always 0, and Rank is 0, 1.

- For each host, assign it the next highest IP address, starting from `<baseip>`
- For each host, use the next MAC address found in the MAC address text file

Running the add-hosts tool

Now that you have a complete XML configuration and MAC address file, you can simply run `add-hosts` to add the hosts to your Platform OCS database:

```
# add-hosts
```

You can also run add-hosts in test mode to verify the your results before actually populating the database and changing any configuration files. To run in test mode:

```
# add-hosts --testmode
```

If a host already exists in the database, the `add-hosts` tool will prompt you to skip the error, skip all subsequent errors, or just abort completely. All messages and errors are logged to `./add-hosts.log`.

Common uses of add-hosts

The following are common uses of the add-hosts tool:

- Adding multiple racks of hosts to the cluster, where each host is named by rack and rank

  We will illustrate this process using an example. Suppose that we have a fictional cluster consisting of 3 racks, with 5 nodes per rack. We want to add all of these nodes to the cluster as compute nodes, with the following hostnames, and IP addresses (assume Class C network):

Node names	IP Addresses
compute-0-0 compute-0-4	192.168.0.100 104
compute-1-0 compute-1-4	192.168.0.105 109
compute-2-0 compute-2-4	192.168.0.110 114

The steps are:

a. Create the MAC address file with the MAC addresses for the hosts above in `/opt/rocks/etc/mac.txt`
b. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` according to our specification:

```
<?xml version="1.0" standalone="yes"?>

<add-hosts>

  <mac_addr_file value = /opt/rocks/etc/mac.txt" />
  <num_hosts_per_rack value = "5" />
  <order_by_rack value = "yes" />
  <netmask value = "255.255.255.0" />

  <subnet>
    <host_prefix value = "compute" />
    <baseip value = "192.168.0.100" />
    <num_hosts_in_subnet value = "15" />
    <appliance value = "compute" />
  </subnet>

</add-hosts>
```

c. Add the hosts to the Platform OCS database:

```
# add-hosts
```

- Adding multiple racks of hosts to the cluster, where each host is named by a single integer value

  We will illustrate this process using an example. Suppose that we have a fictional cluster consisting of three racks, with five nodes per rack. We want to add all of these nodes to the cluster as compute nodes, with the following hostnames, and IP addresses (assume Class C network):

Node names	IP Addresses
compute-0 compute-14	192.168.0.100 114

  The steps are:

  a. Create the MAC address file with the MAC addresses for the hosts above in `/opt/rocks/etc/mac.txt`
  b. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` according to our specification.

  By default, hosts are named by rack and rank. If you want to name the hosts with a single integer, use `<order_by_rack value = "no" />`.

```
<?xml version="1.0" standalone="yes"?>

<add-hosts>

  <mac_addr_file value = "/opt/rocks/etc/mac.txt" />
  <num_hosts_per_rack value = "5" />
  <order_by_rack value = "no" />
  <netmask value = "255.255.255.0" />

  <subnet>
    <host_prefix value = "compute" />
    <baseip value = "192.168.0.100" />
    <num_hosts_in_subnet value = "15" />
    <appliance value = "compute" />
  </subnet>

</add-hosts>
```

  a. Add the hosts to the Platform OCS database:

```
# add-hosts
```

- Adding hosts for different appliance types

  We will illustrate this process using an example. Suppose that we have a fictional cluster consisting of a rack containing 3 nodes we want to configure as LSF HPC master candidate hosts. To get the appliance name, we run insert-ethers to get the list of appliance names. The name of our appliance is "LSF HPC Master".

Node names	IP Addresses
lsfhpc-0-0 lsfhpc-0-2	192.168.0.200 202

  The steps are:

a. Create the MAC address file with the MAC addresses for the hosts above in `/opt/rocks/etc/mac.txt`
b. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` according to our specification:

```
<?xml version="1.0" standalone="yes"?>

<add-hosts>

  <mac_addr_file value = "/opt/rocks/etc/mac.txt" />
  <num_hosts_per_rack value = "5" />
  <order_by_rack value = "yes" />
  <netmask value = "255.255.255.0" />

  <subnet>
    <host_prefix value = "lsfhpc" />
    <baseip value = "192.168.0.200" />
    <num_hosts_in_subnet value = "3" />
    <appliance value = "LSF HPC Master" />
  </subnet>

</add-hosts>
```

c. Add the hosts to the Platform OCS database:

```
# add-hosts
```

- Assigning IP addresses in descending order

We will illustrate this process using an example. Suppose that we have a fictional cluster consisting of 3 racks, with 5 nodes per rack. We want to add all of these nodes to the cluster as compute nodes, with the following hostnames, and IP addresses assigned in reverse order (assume Class C network):

Node names	IP Addresses
compute-0-0 compute-0-4	192.168.0.114 110
compute-1-0 compute-1-4	192.168.0.109 105
compute-2-0 compute-2-4	192.168.0.104 100

The steps are:

a. Create the MAC address file with the MAC addresses for the hosts above in `/opt/rocks/etc/mac.txt`
b. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` according to our specification:

   By default, IP addresses are generated in ascending order. In other words, hosts are assigned the next highest IP address. If you want to generate the addresses in descending order (i.e. assign hosts the next lowest IP address), place `<gen_descending_ip value = "yes" />` in your `<subnet>` section.

```
<?xml version="1.0" standalone="yes"?>

<add-hosts>

  <mac_addr_file value = "/opt/rocks/etc/mac.txt" />
  <num_hosts_per_rack value = "5" />
  <order_by_rack value = "yes" />
  <netmask value = "255.255.255.0" />
```

```
  <subnet>
    <host_prefix value = "compute" />
    <baseip value = "192.168.0.114" />
    <num_hosts_in_subnet value = "15" />
    <appliance value = "compute" />
    <gen_descending_ip value = "yes" />
  </subnet>

</add-hosts>
```

c. Add the hosts to the Platform OCS database:

```
# add-hosts
```

- Assigning a non-continuous series of IP addresses to a rack of hosts

We will illustrate this process using an example. Suppose that we have a fictional cluster consisting of 3 racks, with 5 nodes per rack. We want to add all of these nodes to the cluster as compute nodes, with the following hostnames, and IP addresses (assume Class C network). However, there is an IP address range that cannot be used (192.168.0.105 109).

Node names	IP Addresses
compute-0-0 compute-0-4	192.168.0.100 104
compute-1-0 compute-1-4	192.168.0.110 114
compute-2-0 compute-2-4	192.168.0.115 119

The steps are:

a. Create the MAC address file with the MAC addresses for the hosts above in `/opt/rocks/etc/mac.txt`
b. Create an XML configuration file in `/opt/rocks/etc/add-hostsrc` according to our specification.

To handle two disjoint address ranges, use two separate `<subnet>` sections. One `<subnet>` section defines hosts for the 192.168.0.100 104 range, and the second `<subnet>` section defines hosts for the 192.168.0.110 119 range.

```
<?xml version="1.0" standalone="yes"?>

<add-hosts>

  <mac_addr_file value = "/opt/rocks/etc/mac.txt" />
  <num_hosts_per_rack value = "5" />
  <order_by_rack value = "yes" />
  <netmask value = "255.255.255.0" />

  <subnet>
    <host_prefix value = "compute" />
    <baseip value = "192.168.0.100" />
    <num_hosts_in_subnet value = "5" />
    <appliance value = "compute" />
  </subnet>

  <subnet>
```

```
    <host_prefix value = "compute" />
    <baseip value = "192.168.0.110" />
    <num_hosts_in_subnet value = "10" />
    <appliance value = "compute" />
  </subnet>
</add-hosts>
```

c. Add the hosts to the Platform OCS database:

```
# add-hosts
```

- Replacing a host

Sometimes you might need to physically replace a node in your cluster if it experiences a hardware failure.

If you are replacing a small number of hosts, you can use the `insert-ethers` tool:

 # insert-ethers --replace <*node_name*>

If you are replacing a large number of hosts, you need to do the following:

a. Open `/opt/rocks/etc/mac.txt` and replace the old MAC addresses of the failed hosts with the MAC addresses of the new hosts
b. Remove the hosts corresponding to the removed MAC addresses from the database. To get a mapping of hostnames to MAC addresses, run:

```
# dbreport ethers | grep 00:cc0:9f:45:02:16
00:c0:9f:45:02:16 compute-0-3.local
# insert-ethers -remove "compute-0-3"
```

c. From the command line, run `add-hosts` or `add-hosts --testmode` to test your outcome before making any configuration file changes.

An error will occur indicating trouble adding the first host. This error is expected since you have already added this host to your database. When prompted, tell `add-hosts` **to skip all subsequent errors.**

Check the `./add-hosts.log` **log file to see if you were successful. Each host added successfully has its own line and indicates** `SUCCESS` **at the end. If you failed to add a host, the line indicates** `FAILED`**.**

- Remove a host

The `add-hosts` tool does not provide a way to remove nodes from the Platform OCS cluster. To do this, you need to use the `insert-ethers` tool:

 # insert-ethers --remove <*node_name*>

If you have a large number of hosts to remove, there are some SQL commands you can run to remove a batch of hosts. This is done from the command-line. Here are some ways to do this. Note that there are no line breaks in the commands. All of the commands are typed on one line.

a. Remove an entire rack of hosts.

For example, to remove all of the hosts in rack 0:

```
# echo "delete from nodes where name like 'compute-0-%';" | mysql -u apache
cluster
```

```
# echo "delete from networks where name like 'compute-0-%';" | mysql -u apache
cluster
```

b. Remove all nodes with a particular prefix.

   For example, to remove all compute nodes (that is, all nodes with the compute- prefix):

```
# echo "delete from nodes where name like 'compute-%';" | mysql -u apache
cluster
# echo "delete from networks where name like 'compute-%';" | mysql -u apache
cluster
```

c. Clean out all of the hosts, except for the frontend.

```
# echo "delete from nodes where id != 1;" | mysql -u apache cluster
# echo "delete from networks where node != 1;" | mysql -u apache cluster
```

Other add-hosts documentation

The XML configuration file for add-hosts includes more advance options. For more information about the `add-hosts` tool, refer to the manpage and the `add-hosts` tool section in the Readme document for Platform OCS Rolls:

`http://localhost/homepage/platform/Roll_Readme.html`

Platform OCS installation optimizations

Platform OCS is based on the SDSC Cluster Toolkit 4.1. Platform OCS has been optimized to speed the installation of compute nodes. The original SDSC behavior is still available should it be needed. The optimizations are:

- Single check-in of configuration files. Configuration files generated or altered by Platform OCS would also be checked-in to RCS. The original behavior would check-in the install.log file 100+ times. This change checks in all Platform OCS generated files once during the first boot of a newly installed system. You can revert to the original behaviour using the "logrcs" boot option.
  - For compute nodes, edit the `/tftpboot/pxelinux/pxelinux.cfg/default` file and add `logrcs` to the line that begins with `append`.
  - Frontends will default to the old behaviour.
- Removal of change-logs from XML files. Many of the XML configuration files for SDSC cluster toolkit contained a large proportion of comments. Reading through these comments slows the XML parser. Platform OCS strips the change-logs from the XML files in the rocks-dist. This speeds the process of generating installation configuration files, which allows the frontend to install more hosts at the same time.
- Caching of compute node kickstart file. During installation of compute nodes the frontend must generate an Anaconda kickstart file. This file controls the partitioning, packages, and post installation scripts that are applied to the booting node. Generating the kickstart file was very CPU intensive for the frontend, and is a contributing factor to compute node installation failure. The differences between a kickstart file from one compute node to another is minimal. Platform OCS uses a caching mechanism to reduce the load on the frontend during kickstarting. This allows the frontend to install more compute nodes at the same time. Caching is only used for compute nodes. Other appliances will not use caching. For small clusters caching may not be necessary. Caching may not be compatible with all rolls. It can be disabled by running the following command on the frontend:

  ```
  # touch /home/install/sbin/cache/disable-cache
  ```

- Disabling Bittorrent downloads. SDSC cluster toolkit 4.1 uses bittorrent to transfer files to installing nodes. In tests it was found that installation was actually slower for reinstalling individual nodes, and for installing less than 32 hosts at a time. By default Platform OCS uses http, not bittorrent, to transfer files to installing nodes. A node will require approximately 3-5 seconds of network bandwidth on Gigabit Ethernet. Bittorrent file transfers can be controlled by running the following on the frontend:

```
# /opt/rocks/sbin/rocks-bittorrent {on|off}
```

Warning: If the OCS database is changed manually, it is necessary to remove the kickstart cache file(s). Run:

```
# rm -rf /home/install/sbin/cache/ks.cache*
```

Advanced partitioning

This section describes creating a `replace-auto-partition.xml` file to define a different partitioning layout in compute nodes. By default, Platform OCS only partitions the first disk found on the node. Use this method if you want greater control over the partitioning and/or want to partition multiple disks.

In that section we described the first method, and briefly touched upon the second method. We will discuss the second method in greater detail here. Before we start, be aware that Platform OCS does not support creation of software RAID partitions, or LVM partitions.

Creating a customized partition layout

1. Log into the frontend as root.
2. Create a new `replace-auto-partition.xml` file:

   ```
   # cd /home/install/site-profiles/4.1.1/nodes/
   # cp skeleton.xml replace-auto-partition.xml
   ```

   Remove the <package> and <post> tags from the XML file.

3. The partitioning layout is specified between the <main> and </main> tags. Specify one line for each partition that you want to create or preserve using the <part> and </part> tags. Specify the parameters for the partition between those two tags.

   The syntax for creating a new partition is:

   ```
   <part> mountpoint --size size --ondisk disk [--grow]
   [--fstype] </part>
   ```

   Where:

   - *mountpoint* is the path where the partition is mounted. If you are specifying a swap partition, use `swap`.
   - *size* is the partition size in Megabytes (MB).
   - *disk* is the disk on which to create the partition without the `/dev` prefix. For example, use `--ondisk sdb` to create a partition on the second SCSI disk.
   - The optional `--grow` option tells the installer to use the remaining space to create the partition. When using `--grow`, use `--size 1`.
   - The optional `--fstype` option tells the installer which file system type to use when formatting the partition. If you don't use this option, the default is `ext3` (unless the partition is a swap partition). Valid values are `ext3`, ext2, `vfat`, and `swap`.

   The syntax for preserving an existing partition is:

   ```
   <part> mountpoint --onpart partition [--noformat] </part>
   ```

   Where:

   - *mountpoint* is the same as above

- *partition* is the device name without the `/dev` prefix. For example, use `--onpart sdb1` to preserve the first partition on the second SCSI disk.
- The `--noformat` option tells the installer not to format the partition. By default, all partitions are reformatted. Use `--noformat` if you want to preserve the contents of your partition.

  Example: Install a compute node having 2 SCSI disks using the following partitioning layout:

Mountpoint	Size	Disk	Use Existing Partition
/	30 GB	sda	No
Swap	2 GB	sda	No
/var	10 GB	sda	No
/tmp	10 GB	sda	No
/data1	Rest of Disk	sda	No
/data2	10 GB	sdb	Yes
/data3	Rest of Disk	sdb	No

The `replace-auto-partition.xml` file:

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

<kickstart>

<description>
</description>

<changelog>
</changelog>

<main>
  <part> / --size 30000 --ondisk sda </part>
  <part> swap --size 2000 --ondisk sda </part>
  <part> /var --size 10000 --ondisk sda </part>
  <part> /tmp --size 10000 --ondisk sda </part>
  <part> /data1 --size 1 --grow --ondisk sda </part>
  <part> /data2 --onpart sdb1 --noformat </part>
  <part> /data3 --size 1 --grow --ondisk sdb</part>
</main>

</kickstart>
```

a. Update the Platform OCS distribution

   # **cd /home/install ; rocks-dist dist**

b. Prepare nodes for reinstallation
   i. Remove the `/.rocks-release` file on the compute node you want to reinstall.

      # ssh <*node_name*> **rm -fr /.rocks-release**

   ii. Remove the partitions for the compute node from the Platform OCS database.

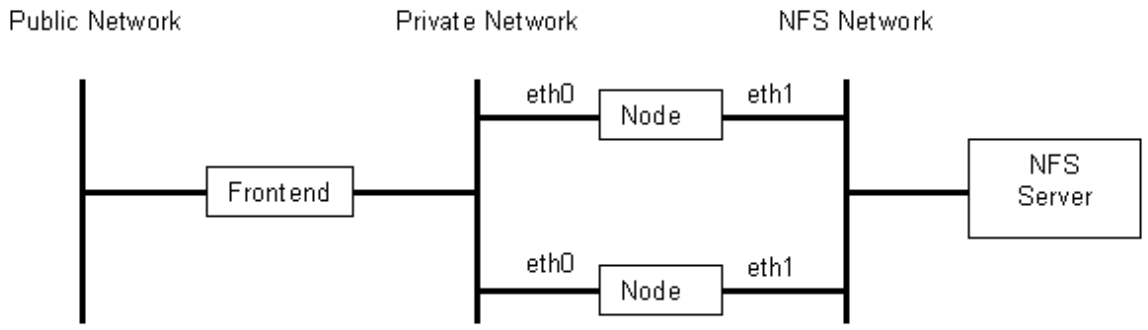
      # **rocks-partition -list -delete -nodename** <*node_name*>

c. Reinstall your nodes

```
# ssh <node_name> '/boot/kickstart/cluster-kickstart'
```

Adding additional network interfaces

Most rack-mount servers have at least two NICs. By default, Platform OCS nodes do not use the second NIC. Other network adaptors can be bonded together, or exclusively used for data. Ethernet bonding can improve the overall bandwidth to each node (up to 30%). Ethernet bonding is not discussed in this document. The diagram below describes a network configuration where one interface (eth0) is connected to the frontend management network and the other interface (eth1) is connected to the NFS server for performance.



Configuring extra NICs is done using the `add-extra-nic` command on the frontend. This command is going to update the networks table in the cluster database. If large numbers of nodes are added it may be quicker to add the extra NICs by directly adding them to the database. The `add-extra-nics` command has the following syntax:

```
# add-extra-nic --if=<nic_interface> --ip=<ip_address> --netmask=<netmask>
--gateway=<gateway> --name=<nic_hostname> --mac=<nic_mac> <node_name>
```

Where:

- <*nice_interface*> Interface is the name of the NIC as it is known on the node, typically eth1.
- <*ip_address*> is the IP address to assign to the NIC.
- <*netmask*> is the network mask for the IP address e.g 255.255.0.0 for a class B
- <*gateway*> (optional) is the IP address of the default gateway for the network that the new NIC is plugged into.
- <*nic_hostname*> is the name to assign to the IP address of the NIC.
- <*nic_mac*> (optional) is the MAC address of the NIC.
- <*node_name*> is name of the node to configure the extra NIC on.

For example:

```
# add-extra-nic --if=eth1 --ip=192.168.1.1 --netmask=255.255.0.0
--name=nasnet-0-0 compute-0-0
```

This configures eth1 on compute-0-0 with IP address 192.168.1.1/16 and assigns the interface the name `nasnet-0-0`.

Note: Nodes should be booted once before running add-extra-nic so that the MAC address of the extra NIC card(s) are added to the database. If a NIC card is installed on the machine after running add-extra-nic the database will contain incorrect entries for the NIC

If you have installed a NIC card after running `add-extra-nic` then you will need to edit the database to update the MAC and IP entries for the NIC. The procedure below outlines how to correct the bad entry:

1. Connect to the database.
2. Browse the node table in the database to determine the ID of the node with the extra NIC.
3. Examine the network table using the node ID to locate the entries for the node.
4. One entry will have the IP address of the extra NIC, the other will have a MAC address, but no IP address. These two entries need to be merged into one.
5. Make note of the MAC, Module, then edit the entry with the IP address, and update the MAC and module.
6. Delete the entry that had a MAC, but no IP.
7. Reinstall the node.

Other useful options for "add-extra-nic" are:

--dryrun

This will produce the SQL code corresponding to adding the NIC, for example:

```
# add-extra-nic --if=eth1 --ip=192.168.1.1 --netmask=255.255.0.0
--name=nasnet-0-0 --dryrun compute-0-0 query is: update networks set
ip="192.168.1.1" ,netmask="255.255.0.0" ,gateway=NULL ,name="nasnet-0-0"
,mac=NULL ,module=NULL where node=2 and device="eth1"
```

This is useful for creating the SQL for adding many nodes at once, however care must be taken to get the node number correct.

--module

This is the name of the kernel module for the NIC card.

Kickstart/PXE bootstrap process

The Platform OCS Kickstart installation process requires DHCP, TFTP, Web, and CGI scripts on the frontend to complete installation of a node. What follows will be a brief overview of the DHCP, and TFTP services, followed by a description of the generation of the configuration file for the node installer, and then a summary of the entire installation process.

DHCP service

Dynamic Host Configuration Protocol (DHCP) is a protocol whereby a booting node can gather sufficient information to connect to a network. In Platform OCS it provides the node with the following:

- An IP address
- A subnet mask
- A broadcast address
- A default router
- A DNS domain
- An NIS domain
- The next server to get information from.

The booting node uses the DHCP protocol to get enough information to connect to the network and to know which server to connect to next.

TFTP service

The Trivial File Transfer Protocol (TFTP) service is used to download files from the TFTP server. It can be thought of as a simpler version of FTP, but with no passwords. Installing nodes use this protocol to download the Linux kernel and root filesystem for the installer.

Kickstart generation

Platform OCS uses a modified Redhat Anaconda installer to perform the installation. It has been extended to allow a remote console on the installing nodes, and to permit post-installation scripts. Anaconda uses a Kickstart file to direct its operation. The kickstart file tells Anaconda:

- What RPMs to install.
- What partitioning scheme to use.
- Root password.
- Timezone and Locale.
- What post-install scripts to run.

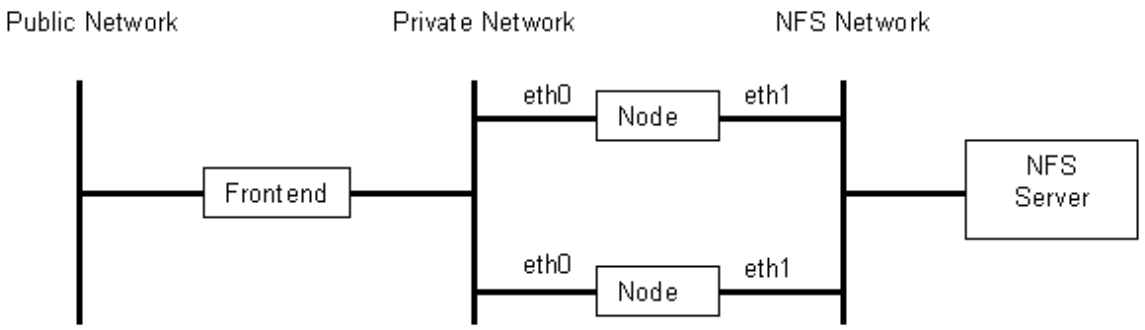
Platform OCS packages pre-configured applications called rolls. The rolls contain:

- The RPMs to install
- The post installation scripts to configure the roll.
- The installation hierarchy i.e. what pieces to install where.

Consider the PVFS2 roll. This roll contains RPMs for the PVFS2 applications and source code. It also contains scripts to build the kernel driver and information describing the components to install on the different PVFS2 node types.

To install a node a kickstart file that is appropriate for that node type has to be generated. This is known as the kickstart generation process. The resulting kickstart file must direct Anaconda to install the appropriate RPMs and run the correct post-installation scripts. Platform OCS nodes automatically generate the kickstart file during installation. The Kickstart generation tools can also be run manually for debugging.

As of Platform OCS 4.1.1 the kickstart file is no longer generated completely by the frontend. The diagram below illustrates how the kickstart file is generated.



1. The installing node requests an intermediate kickstart file through a secure web request to URL: `https://frontend/install/sbin/public/kickstart.cgi`.
2. The kickstart.cgi script runs and queries the database for information about the node it is kickstarting.
3. The `kickstart.cgi` calls kpp.
4. The Kickstart PreProcessor (kpp) runs and extracts the list of RPMs and post-install scripts from the rolls to install, based on the appliance type it has to install. It returns the result to `kickstart.cgi`.
5. The `kickstart.cgi` script encapsulates the kpp output in html and returns the result to the installing node.
6. The installing node runs the Kickstart Generator (kgen) to generate the final kickstart file for Anaconda to install with.

The resulting anaconda kickstart file lists all the RPMs needed for the type of appliance the node should be, as well as the post-installation scripts to configure the rolls it contains.

PXE bootstrap process

Installing nodes use Pre-eXecution Environment (PXE) to initiate the installation procedure. Using PXE the booting node will gather such information as:

- Network information (through DHCP).
- Server to get booting information from.
- Boot configuration, i.e. kernel name, and kernel options.

The following explains the sequence of events a configured node will go through to boot. An understanding of the process will help troubleshoot installation failures.

1. When the node is powered up it will attempt to boot from the list of boot devices. If the node was shutdown hard, or it has a new disk, booting from the local disk will fail.
2. The node will then attempt to boot from the network devices.
3. The node will request an IP address.
4. If the node was added to the frontend either through insert-ethers, or add-hosts, the frontend will respond to the DHCP request with an IP address, subnet mask, default router, and the IP address of the next server to get booting info from. If the frontend does not know the node, the frontend will ignore its DHCP requests. This is why you will see unanswered DHCP requests in the logs.
5. The node will them contact the frontend to get the boot information. During bootup you will see something like:

```
Trying to load: pxelinux.cfg/00-11-22-33-44-55
Trying to load: pxelinux.cfg/0AFFFFFE
Trying to load: pxelinux.cfg/0AFFFFF
Trying to load: pxelinux.cfg/0AFFFF
Trying to load: pxelinux.cfg/0AFFF
Trying to load: pxelinux.cfg/0AFF
Trying to load: pxelinux.cfg/0AF
Trying to load: pxelinux.cfg/0A
Trying to load: pxelinux.cfg/0
Trying to load: pxelinux.cfg/default
```

6. The node will use TFTP to retrieve the configuration file `/tftpboot/pxelinux/pxelinux.cfg/default`. This file tells the node which kernel and initial RAM disk (initrd) to download next.
7. The node uses TFTP to download the kernel and initrd.
8. The node runs the kernel with the arguments supplied in the configuration file.
9. The Linux kernel starts booting and runs the installation program on the initrd.
10. The installation program requests the kickstart of configuration files for the installer. This is the blue screen with the Secure Kickstart dialog. An intermediate kickstart file is generated by the frontend and sent to the installing node. The installing node uses kgen to generate the final kickstart file. The resulting kickstart file can be found on the installing node in: `/tmp/ks.cfg`
11. The installed then uses the HTTP protocol to retrieve a series of files. They are:
    - `updates.img`
    - `product.img`
    - `netstg2.img`
12. It starts the real Anaconda installer.
13. The real anaconda retrieves further files using the HTTP protocol. They are:
    - `hdlist`
    - `comps.xml`
    - `hdlist2`
    - `comps.rpm`
14. Anaconda partitions and formats the first disk.
15. It then starts downloading and installing the RPM packages.
16. It installs the boot loader.
17. It then runs the post-installation scripts provided by the rolls.
18. The node reboots.
19. During the first reboot the node may run a series of scripts to build kernel modules and finish roll installation.

Troubleshooting the installation process

This section presents some hints on how to troubleshoot the installation process. Use the bootstrap section above to first locate which services/processes are at play.

DHCP service

In Platform OCS only nodes known to the frontend can get a successful response from the DHCP service, so a DHCP timeout on the installing node is not always an indication of a DHCP failure. It could simply mean that the node has not been added to the frontend either by insert-ethers or add-hosts. Other reasons for failure are:

- Network congestion. The DHCP request is a UDP packet that may get dropped by the switches.
- Broadcast suppression. Some switches have a feature to suppress broadcasts. If this is turned on the DHCP requests may not reach the frontend.
- DHCP server host (frontend) too busy. During installation the frontend may be under a very high load, especially if many hosts are installing at a time. It may not respond to the nodes request for an IP in a timely manner. This is particularly true when the node is being installed for the first time.

TFTP service

A failure of the TFTP service will manifest its self after the DHCP response, while the node is trying to download the `pxelinux.cfg/default`, `kernel`, `initrd`. The files the frontend are serving are located in `/tftpboot/pxelinux`. Nodes will not be able to retrieve files outside this directory. In Platform OCS the xinetd service is responsible for starting the TFTP service. The xinetd process must be running on the frontend to start the TFTP service when needed. Use the following command to verify xinetd is running, and configured to start TFTP

```
# netstat -ap |grep tftp
```

You should see a UDP listner, and the PID of the xinetd process.

Kickstarting

The kickstarting process requires several components to work correctly to generate a kickstart file. The first step in testing kickstarting is to test if an intermediate kickstart file can be generated. The following commands test kickstart generation:

```
# cd /home/install/sbin
# ./kickstart.cgi -c <node_name> > ks.out
```

Check the output of the cached Kickstart file with the following commands:

```
# cd /home/install/sbin
# ./kickstart.cgi -c <node_name> --usecache=1 > ks.out
```

Note the Kickstart and cached Kickstart files will differ because the SSL certificates are regenerated each time. The resulting file should be over 200Kbytes. If it fails completely check that the database is running by running:

```
# mysql -u apache cluster
mysql> quit;
```

You should be able to connect to the database. If that fails try restarting the mysqld service.

If the file is too small, look at the first few lines of the file. If you see something like:

```
Status: 503 Service Busy
Content-type: text/html
<hl>Service is Busy<hl>
```

either the cache file is corrupt, or the lock file is out of sync. If the `--usecache=1` argument was used then the

cache file is corrupt. The kickstart cache file is `/home/install/sbin/cache/ks.cache.<architecture>`. If the file is corrupt simply delete it, and a new one will be generated the next time a compute node attempts to reinstall. If the `-usecache=1` option was not used, then check for other running kickstart.cgi scripts. If there are no other running scripts, delete the file `/var/tmp/kickstart.cgi.lck`.

411

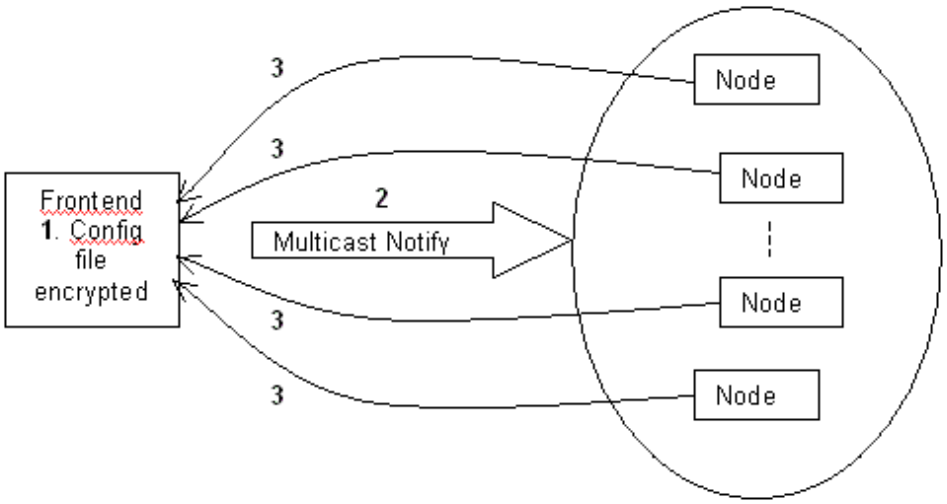
The 411 service is responsible for synchronizing common configuration files across the cluster. It acts as a secure replacement for NIS in Platform OCS clusters. In Platform OCS 4.1.1 it is used to distribute:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`
- `/etc/services`
- `/etc/rpc`
- `/etc/auto.*`

This section will discuss common tasks with 411 such as:

- Triggering a file update on all nodes.
- Transferring any configuration file to all nodes.
- Adding a configuration file to the 411 make file.
- Retrieving 411 files on a compute node.
- Changing the polling rate for file retrieval.
- Distributing different files to different nodes.

Before discussing the common 411 tasks it is necessary to discuss how 411 works. This will help in understanding 411's behavior. 411 uses multicasts to alert nodes that files have changed. The nodes have a greceptor process listening for 411 broadcasts. When a broadcast is received the greceptor process will retrieve the file(s) that have changed. In the event that the broadcast message is lost, or the intermediate networking devices do not forward multicasts the process will also poll the frontend periodically and retrieve all the files in the `/etc/411.d` directory. The diagram below illustrates the sequence of events.



1. The configuration file(s) that have changed are encrypted using PGP, and stored in `/etc/411.d`.
2. A multicast broadcast is sent out the private interface of the frontend. The multicast contains a list of files to update. The nodes are running the greceptor service which is listening for multicasts.
3. The nodes that received the multicast use HTTP to retrieve the changed file(s). The files are decrypted and wrote to the desired location.

Triggering a file update on all nodes

When a file is changed on the frontend the administrator must instruct 411 to notify all nodes in the cluster that a configuration file has changed. This is done by running:

```
# make -C /var/411
```

This command will encrypt the files that have changed using PGP, store the resulting file in `/etc/411.d`, and then send a multicast to all nodes alerting them of to the changed file. The nodes will then retrieve just those files that have changed.

It is not always necessary to run the command above when files are changed. Some commands such as the "useradd" script will automatically update the passwd, shadow, group, and auto.home files then instruct 411 to update the files on the nodes. To force all files to be updated use:

```
# make -C /var/411 force
```

Warning: Using the command above if any files were distributed using 411put (see below), they will no longer be available, however all the nodes will try to download them every 30 seconds.

Transferring any configuration file to all nodes:

It is possible to transfer any configuration file to all nodes. This is done by running:

```
# 411put <fully_qualified_file_name>
```

This will encrypt the file, store it in /etc/411.d and notify the nodes. The nodes will retrieve the file either when the notification is received, or periodically. If the file is changed on the frontend you must run 411put each time it changes. Running `make -C /var/411`" will not update a file transferred with 411put..

Warning: Files added this way will not be available to the nodes after using "make -C /var/411 force". This is because the contents of /etc/411.d are removed.

Adding a configuration file to the 411 makefile:

If you add files to the 411 makefile instead of using 411put then the file will be automatically distributed to the nodes when `make -C /var/411` is run. To make this work it is necessary to do the following:

1. Edit `/var/411/Files.mk` and add an entry for the file you wish to update like:

   ```
   FILES += /etc/myfile
   ```

2. Now regenerate the `411.mk` file by running:

   ```
   # make -C /var/411 411.mk
   ```

3. Finally send a notification that the files have changed by running:

   ```
   # make -C /var/411 force
   ```

Retrieving 411 files on a compute node:

On a node it is possible to retrieve files from the frontend using 411. The node will retrieve the latest file from the frontend, made available with `make -C /var/411`, or `411put`. If you change a 411 shared file on the frontend and do not run `make -C /var/411` then the latest version of the file will not be distributed to the nodes. Only those files

that have been made available through 411 can be retrieved. A list of available files can be seen by looking at `http://fontend/411.d/`

Note: File names have "`/`" replaced with "`.`", and "`.`" replaced with "`..`".

Use the `411get` command on the node to retrieve a file from the frontend, for example

```
# 411get /etc.passwd
```

This will get a copy of the /etc/passwd file from the frontend, alternatively on the node run:

```
# 411get
```

This will get a list of all available files.

Changing the polling rate for 411 file retrieval

The multicast broadcasts the frontend uses to notify the nodes that files have changed are not guaranteed to be received by the nodes either because of:

- Mis-configured switches/routers.
- Network congestion.
- Nodes down

To work-around this problem the nodes periodically poll the frontend, however the default poll rate is every 5 hours. The SDSC documentation discusses how to change this value, however it is incorrect Editing the `/etc/411.conf` file does not work. In fact editing the `411.conf` file will cause the file updates to fail. The workaround is:

1. Edit the `/opt/ganglia/lib/python/metrics/event411.py` file on the node you wish to change.
2. Locate the `self.interval = 18000` line in the `__init__` method and change it to the desired value. Note the timing is poor, and values less than 300 do not work.
3. Save the file.
4. Run the following commands:

```
# service greceptor stop
# service greceptor start
```

Distributing different files to different nodes

It is possible to use 411 to distribute different files to different nodes in the cluster. This is done using 411 groups. Nodes will listen for and download those files that are part of the group they belong to. A nodes group membership is controlled by the `/etc/411.conf` file. The example below is for a node which is part of the `Compute` and `MyGroup/app1conf` group.

```
<!-- Configuration file for the 411 Information Service -->
<config>
  <master url="http://10.1.1.1/411.d/"/>
  <group>Compute</group>
  <group>MyGroup/app1conf</group>
</config>
```

The `Compute` group is the default for all nodes and will provide the following:

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`

- /etc/services
- /etc/rpc
- /etc/auto.*

The MyGroup/app1conf is a nested group. It will include all the files from the 411 group MyGroup and all of the files in MyGroup/app1**.** A node that was part of just the MyGroup would only get the files from MyGroup and not from MyGroup/app1.

Adding a 411 group

Adding a 411 group requires changes on both the frontend, and the nodes that will be part of the group. This section covers how to add a new group, starting with the change necessary on the nodes.

1. Edit the /etc/411.conf file
2. Add a group entry after the Compute group, for example,

   ```
   <group>MyGroup</group>
   ```

3. Restart the greceptor service using:

   ```
   # service greceptor restart
   ```

This needs to be done on all nodes that are to be part of the MyGroup 411 group. On the frontend the following steps have to be performed:

1. Make the directory: /var/411/groups/MyGroup/

   ```
   # mkdir /var/411/groups/MyGroup
   ```

2. Place the files you want to distribute in the above directory. The path on the node will be relative to the /var/411/groups/MyGroup/ directory. If a node is to have the file /etc/appfile1, then on the frontend it should be placed in /var/411/groups/MyGroup/etc/appfile1, for example:

   ```
   # mkdir /var/411/groups/MyGroup/etc
   # cp <path_to_appfile1> /var/411/groups/MyGroup/etc/appfile1
   ```

3. Edit the file /var/411/Group.mk.
4. Change the value of GROUPS e.g.

   ```
   #GROUPS = Storage-Node Math-Node
   GROUPS = MyGroup
   ```

5. Add the code below to the bottom of the file.

Note: Make files, such as this, require Tab characters, not spaces.

```
MYGROUP_FILES = \
    /var/411/groups/MyGroup/etc/appfile1 \
    /var/411/groups/MyGroup/etc/appfile2

MyGroup:
    @echo "Rebuilding 411 Group makefile for $@..."
    @files="$(MYGROUP_FILES)"; \
    echo "## $@ Group" >> 411-Group.mk; \
    echo >> 411-Group.mk; \
```

```
    echo -n "all: " >> 411-Group.mk; \
    for f in $$files; do \
        echo -n "`$(PUT) --411name --chroot=/var/411/groups/$@ \
            --group=$@ $$f` "; \
    done >> 411-Group.mk; \
    echo >> 411-Group.mk; \
    echo >> 411-Group.mk; \
    for f in $$files; do \
        echo "`$(PUT) --411name --chroot=/var/411/groups/$@ \
            --group=$@ $$f`:: $$f"; \
        echo "  $(PUT) --group=$@ --chroot=/var/411/groups/$@ \$$?"; \
        echo; \
    done >> 411-Group.mk
```

This line has a Tab character after the double quotes, not spaces

---

6. Edit the MYGROUP_FILES list to include the list of files to copy over. Remember to end each line with a "\", unless it is the last file in the list.
7. Run the command below:

   ```
   # make -C /var/411 groups
   ```

8. You will see something like:

   ```
   make: Entering directory `/var/411'
   make MyGroup
   make[1]: Entering directory `/var/411'
   Rebuilding 411 Group makefile for MyGroup...
   make[1]: Leaving directory `/var/411'
   make: Leaving directory `/var/411'
   ```

9. Now trigger the update by running:

   ```
   # make -C /var/411
   ```

10. For each file added you should see something like:

    ```
    /opt/rocks/sbin/411put --group=MyGroup --chroot=/var/411/groups/MyGroup
    /var/411/groups/MyGroup/etc/appfile1
    411 Wrote: /etc/411.d/MyGroup/etc.appfile1
    Size: 2512/1681 bytes (encrypted/plain)
    Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1
    ```

11. If you do not see a 411put for each of the files either:
    - The file has already been sent.
    - The `Group.mk` file has spaces where there should be Tabs.
12. Verify that the node has the files.

Adding a nested 411 group

Adding a nested 411 group is very similar to the previous instructions. On the nodes:

1. Edit the `/etc/411.conf` file.
2. Add the nested group entry after the Compute group. Note it is not necessary to specify the parent group in a separate line. It is already implied, that is, only add:

```
<group>MyGroup/app1conf</group>
```

3. Restart the greceptor service using:

```
# service greceptor restart
```

This needs to be done on all nodes that are to be part of the `MyGroup/app1conf 411` group. On the frontend the following steps have to be performed:

1. Make the directory: `/var/411/groups/MyGroup/`

```
# mkdir /var/411/groups/MyGroup/app1conf
```

2. Place the files you want to distribute in the `/var/411/groups/MyGroup/app1conf` directory. The path on the node will be relative to the `/var/411/groups/MyGroup/app1conf` directory. If a node is to have the file `/etc/otherfile1`, then on the frontend it should be placed in: `/var/411/groups/MyGroup/app1conf/etc/otherfile1`, for example,

```
# mkdir /var/411/groups/MyGroup/app1conf/etc
# cp otherfile1 /var/411/groups/MyGroup/app1conf/etc/otherfile1
```

3. Edit the file `/var/411/Group.mk`.
4. Change the value of `GROUPS`, for example,

```
#GROUPS = Storage-Node Math-Node
GROUPS = MyGroup MyGroup/app1conf
```

5. Add the code below to the bottom of the file.

Note: Make files, such as this, require Tab characters not spaces.

```
NESTEDGROUP_FILES = \
    /var/411/groups/MyGroup/app1conf/etc/otherfile1 \
    /var/411/groups/MyGroup/app1conf/etc/otherfile2

MyGroup/app1conf:
    @echo "Rebuilding 411 Group makefile for $@..."
    @files="$(NESTEDGROUP_FILES)"; \
    echo "## $@ Group" >> 411-Group.mk; \
    echo >> 411-Group.mk; \
    echo -n "all: " >> 411-Group.mk; \
    for f in $$files; do \
        echo -n "`$(PUT) --411name --chroot=/var/411/groups/$@ \
            --group=$@ $$f` "; \
    done >> 411-Group.mk; \
    echo >> 411-Group.mk; \
    echo >> 411-Group.mk; \
    for f in $$files; do \
        echo "`$(PUT) --411name --chroot=/var/411/groups/$@ \
            --group=$@ $$f`:: $$f"; \
        echo "  $(PUT) --group=$@ --chroot=/var/411/groups/$@ \$$?"; \
        echo; \
    done >> 411-Group.mk
```

This line has a Tab character after the double quotes, not spaces

6. Edit the NESTEDGROUP_FILES list to include the list of files to copy over. Remember to end each line with a "\", unless it is the last file in the list.
7. Run the command below to

```
# make -C /var/411 groups
```

8. You will see something like:

```
make: Entering directory `/var/411'
make MyGroup MyGroup/other
make[1]: Entering directory `/var/411'
Rebuilding 411 Group makefile for MyGroup...
Rebuilding 411 Group makefile for MyGroup/app1conf...
make[1]: Leaving directory `/var/411'
make: Leaving directory `/var/411'
```

9. Now trigger the update by running:

```
# make -C /var/411
```

10. For each file added you should see something like:

```
/opt/rocks/sbin/411put --group=MyGroup/app1conf
--chroot=/var/411/groups/MyGroup/app1conf
/var/411/groups/MyGroup/app1conf/etc/other1
411 Wrote: /etc/411.d/MyGroup/app1conf/etc.other1
Size: 295/44 bytes (encrypted/plain)
Alert: sent on channel 255.255.255.255:8649 with master 10.1.1.1
```

11. Verify that the node has the files.

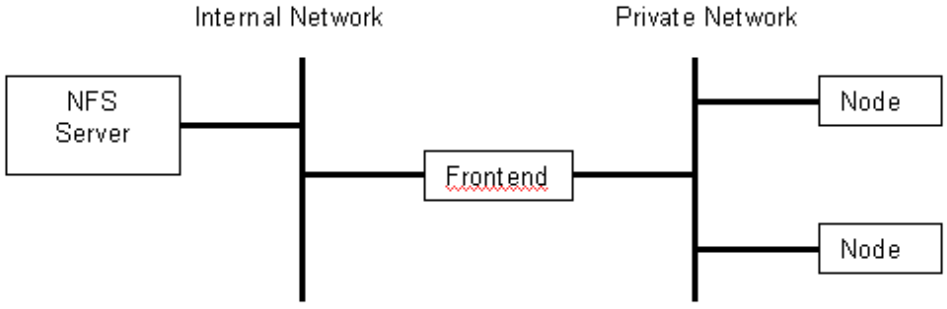
IP tables and routing

This topic covers the following advanced routing and firewall configurations:

- Accessing an NFS server outside the cluster
- Adding other routes on the nodes
- Using a real router instead of the frontend

Accessing an NFS server outside the cluster

Configuring nodes to access existing NFS servers that are on the public network is possible, however it typically requires changes to the intermediate routers. Consider the simple configuration below:



The frontend will be able to mount file systems from the NFS server on the Internal Network, however the nodes on the private network typically will not be able to. The routing tables on the nodes have the frontend as the default gateway, which is correct. However the NFS server will probably not know which router to send to to reach the private network. The issue can be resolved either by:

- Updating the routing table on the NFS server to use the frontend as the gateway to the private network. Consult the administration guide, or man pages, for the NFS server on how to alter the routing table.
- Update the configuration on the main router on the Internal network to direct traffic destined for the private network to the frontend. Consult the user guides for your networking equipment on how to do this.
- Add another NIC to the NFS server and connect that NIC to the Private Network. This solution can provide the best performance.

Adding other routes on the nodes

It is possible to add other routes to the nodes, however all nodes will receive the same routing configuration.

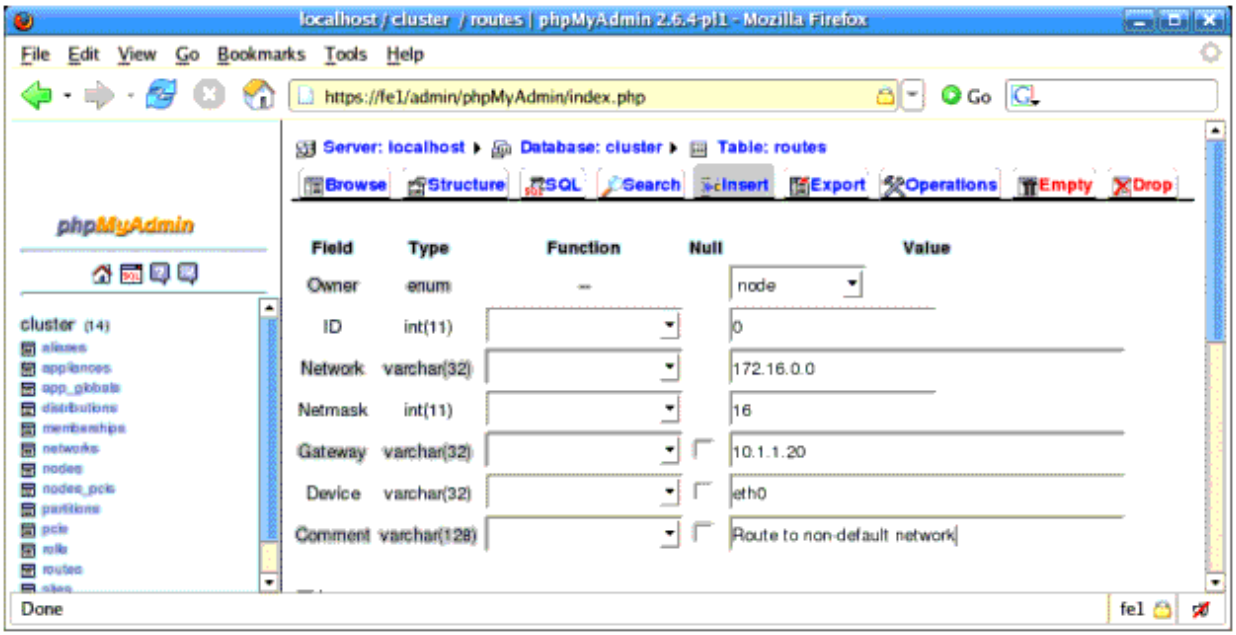
Note: If the route is for a secondary NIC then all nodes should have a second NIC.

Additional routes are added/modified by updating the database on the frontend. The procedure below explains how to add a new route:

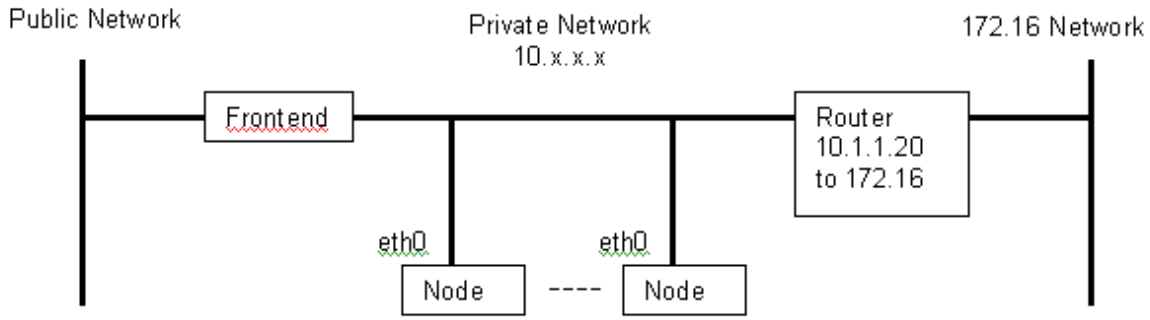
1. Start a web browser on the frontend and go to:

   ```
   http://localhost/admin/phpMy/Admin/index.php
   ```

2. Enter the root login and password.
3. Click on the **routes** link on the left pane.
4. Click on the **Insert** tab at the top.
5. The Insert screen will appear as below:



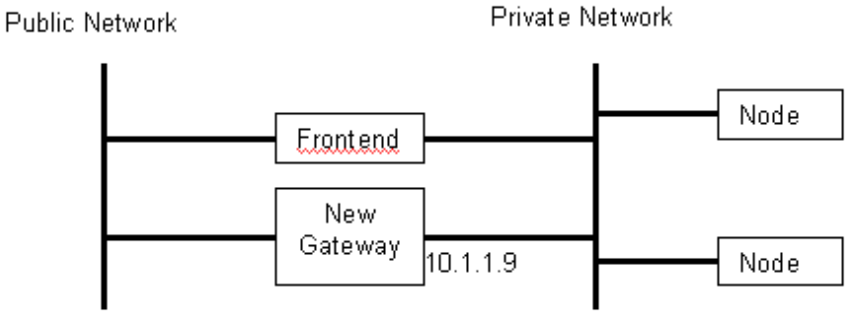
6. Add the routing information for your network. Once added click on the **Go** button at the bottom of the screen. In the example above, the route is for a network structured as seen below:



7. Click on the browse tab and verify the route has been added.
8. On the frontend delete: `/home/install/sbin/cache/ks.cache.x86_64`
9. Reinstall one node to test with.
10. When the node finishes reinstalling verify that the route works.
11. Reinstall all remaining nodes.

Using a real router instead of the frontend

The frontend acts as a router and firewall for the nodes connected to its private interface. This can place a heavy load on the frontend, and degrade user experience. It is possible to use a hardware device to route between the public and private networks. This section discusses how to set the nodes to route correctly through the new gateway. The diagram below illustrates the configuration.



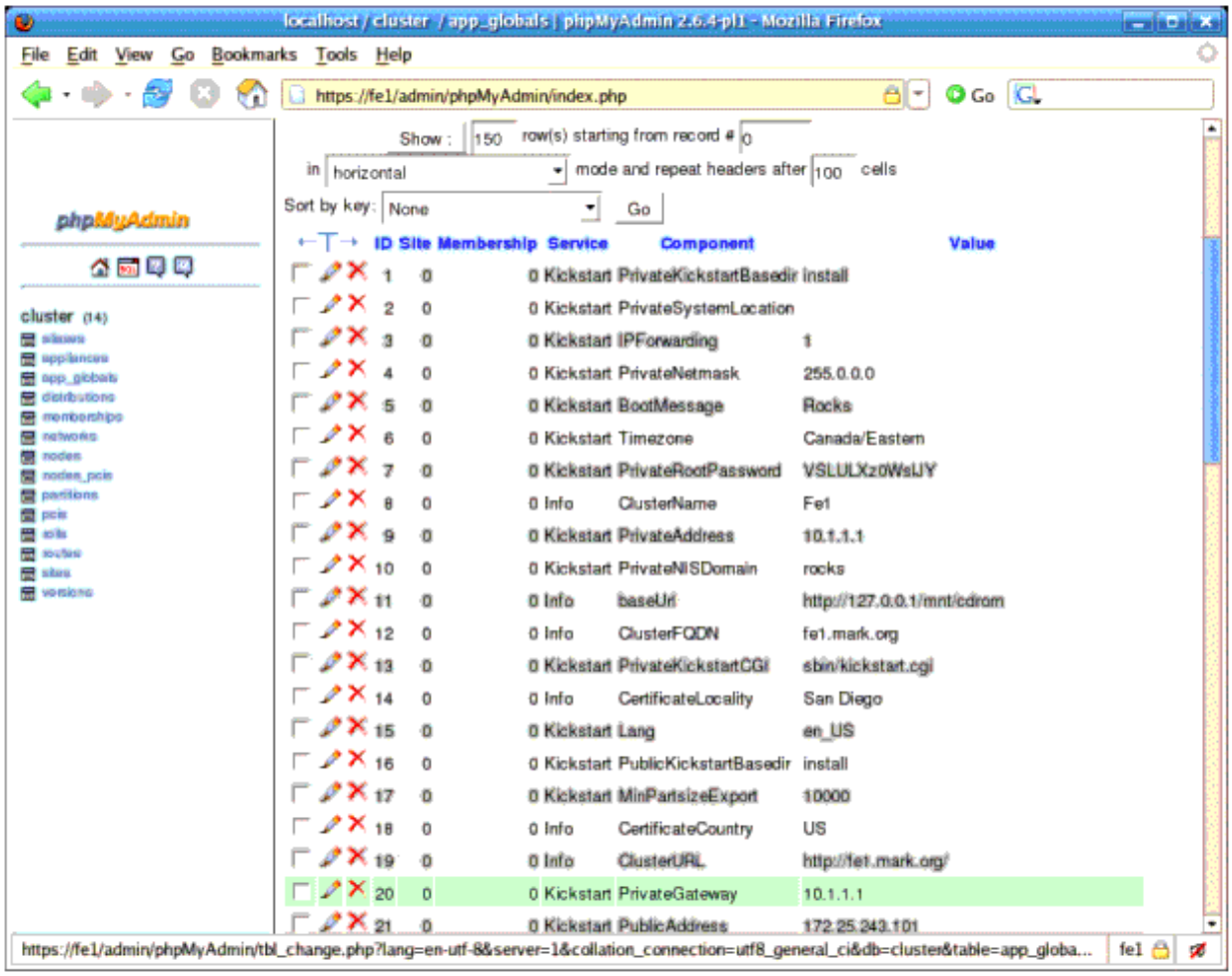
Note: The frontend is a firewall for the nodes. If a router is used, the nodes will be exposed to malicious traffic.

The following procedure explains how to change the network configuration to the one depicted above:

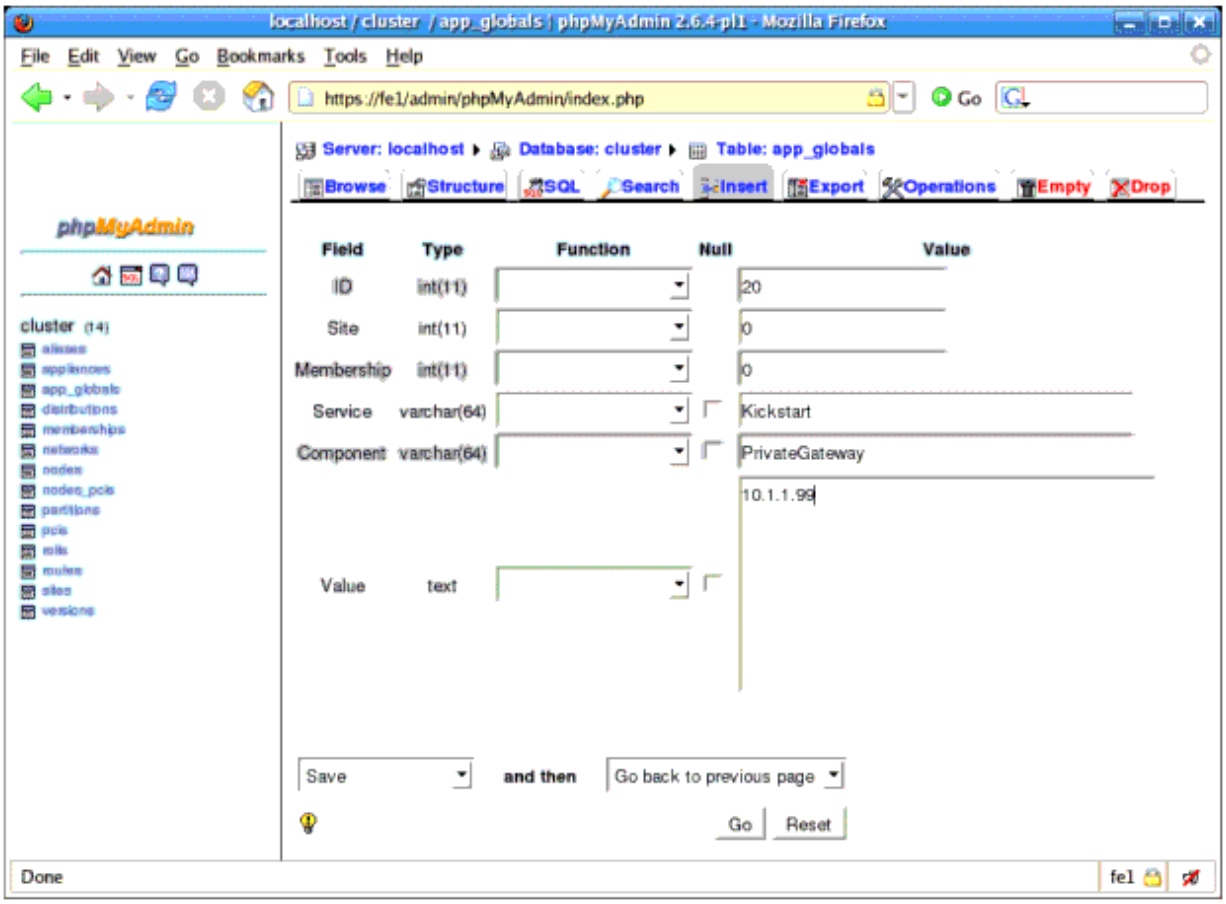
1. Start a web browser on the frontend and go to:

   `http://localhost/admin/phpMy/Admin/index.php`

2. Enter the root login and password.
3. Click on the **app_globals** link on the left side. Then click on the **Browse** tab at the top. You will see something like this:



4. Locate the PrivateGateway entry (seen highlighted in green above), and click on the pencil icon on the left to edit the entry.
5. A screen like the one below will appear.



6. Edit the value, and set it to the IP address of the routers private interface.
7. On the frontend delete: `/home/install/sbin/cache/ks.cache.x86_64`
8. Reinstall one node to test with.
9. When the node finishes reinstalling verify that the route works.
10. Reinstall all remaining nodes.

Mysql database

Platform OCS uses a MySql database to store cluster configuration information. This section will provide an overview of the database tables, review the backup procedure, as well as how to restore the database.

The database has the following tables:

- aliases
- app_globals
- appliances
- distributions
- memberships
- networks
- nodes
- nodes_pcis
- partitions
- pcis
- rolls
- routes
- sites
- versions

The NPACI documentation contains more information about the database schema.

The database is backed-up daily at 4:02 am by a cron job. The actual script used is `/etc/cron.daily/backup-cluster-db`. It uses the mysqldump command to dump the cluster database and store it in the file: `/var/db/mysql-backup-cluster`. It them checks this file into RCS. The backup script can be run manually by root, hocustom reportwever RCS will only update if there has been a change. To see when the database has changed the following commands can be used:

```
# cd /var/db
# rlog mysql-backup-cluster
```

This will produce a list of the RCS check-ins, so it is possible to revert back to a prior version by checking out a previous version and restoring that. To us a previous revision use the command above to locate the database revision you wish to use then run:

```
# cd /var/db
# co -r <revision> mysql-backup-cluster
```

To check-out the backup file. Once the file has checked-out the database can be restored using:

```
# mysql cluster < mysql-backup-cluster
```

or

```
# mysql -e "source <full_name_of_mysql-backup-cluster>"
cluster
```

Custom kernel installation

This section outlines how to change the kernel on the nodes. Changing to a kernel other than the one provided should be avoided if possible. The following section outlines how to install a kernel from `kernel.org`

Warning: The procedure below will change the kernel version to one not supported by RedHat or CentOS, and may make getting support from them difficult. Carefully consider the ramifications of changing the kernel before doing so.

---

1. Go to http://www.kernel.org/pub/linux/kernel/v2.6 to download a kernel you want.
2. Extract the `tar.gz` or `tar.bz2` file to a location of your choice.
3. Copy the existing Red Hat config file from `/boot` directory to the location you extracted the kernel sources to.
4. Go to the directory containing the kernel source.
5. `# make defconfig`
6. `# make rpm`
7. This will produce an RPM from the kernel.

Note: If you want to build a SMP kernel you need to build this on a SMP machine likewise for a UP kernel.

---

8. As root backup the kernel RPM from `/home/install/ftp.rocksclusters.org/pub/rocks/rocks-4.1.1/rocks-dist/rolls/os/4.1.1/x86_64/RedHat/RPMS`.
9. Copy the new kernel RPM to the location above.
10. `# cd /export/home/install`
11. Run `rocks-dist dist`
12. Install one node and verify it has the new kernel.
13. Boot the reinstalled node. If needed edit the `/boot/grub/rocks.conf` and `/boot/grub/grub-orig.conf` to boot the newly installed kernel, and reboot again.
14. Once the reinstalled node is functioning properly reinstall the other nodes.

[Top]

Get Technical Support

Contact Platform

Contact Platform Computing or your Platform OCS vendor for technical support. Use one of the following to contact Platform technical support:

Email

support@platform.com

World Wide Web

http://www.platform.com

Mail

Platform Support
Platform Computing Corporation
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7

When contacting Platform, please include the full name of your company.

See the Platform Web site at http://www.platform.com/Company/Contact.Us.htm for other contact information.

Get patch updates and hotfixes

Obtain the latest patches and hotfixes for Platform OCS from the following page:

http://my.platform.com/products/platform-ocs

To obtain a user name and password, contact Platform Computing technical support at support@platform.com

[Top]

Copyright and Trademarks

Although the information in this document has been reviewed, Platform Computing Corporation ("Platform") does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO

ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

Trademarks

This product includes software developed by the Rocks Cluster Group at the San Diego Supercomputer Center at the University of California, San Diego and its contributors.

® LSF and LSF HPC are trademarks or registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

™ ACCELERATING INTELLIGENCE, PLATFORM COMPUTING, and the PLATFORM and Platform OCS logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

® Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and.or other countries.

® Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

[Top]

Date Modified: August 03, 2006
Platform Computing: www.platform.com

Platform Support: support@platform.com
Platform Information Development: doc@platform.com