Sun™ HPC Software, Linux Edition 1.1

Installation Guide

Sun Microsystems, Inc. www.sun.com

Part No. 820-6354-10

November 2008

Table of Contents

Introduction

Sun™ HPC Software, Linux Edition ("Sun HPC Software") is an integrated open-source software solution for Linux-based HPC clusters running on Sun hardware. It provides a framework of software components to simplify the process of deploying and managing large-scale Linux HPC clusters.

Software Components

The Sun HPC Software provides software to provision, manage, monitor and operate large scale Linux HPC clusters. It serves as a foundation for optional add-ons such as schedulers and other components not included with the solution.

All components included in the Sun HPC Software stack are open source software. See Appendix A for descriptions of the components. For information about the license under which each component is distributed, see the component website.

System Requirements

The table below shows the Sun HPC Software system requirements.

Platform	Sun x64
Operating System	Red Hat Enterprise Linux 5.2 CentOS 5.2 x86_64
Networking	Ethernet, InfiniBand

Audience

This installation guide is written for administrators with a basic knowledge of Linux, familiarity with cluster systems and networking, and some experience with provisioning and configuration tools, such as oneSIS or Cobbler, and system management tools, such as PowerMan or ConMan.

Installing the Sun HPC Software

Completing the installation procedure described in this guide installs the Sun HPC Software on a cluster configured similar to that shown in Figure 1. This example cluster contains:

- **Head Node -** As part of the Sun HPC Software installation process, the Cobbler and oneSIS provisioning tools are installed on the head node. These tools are used for the provisioning of diskful and diskless cluster nodes. The head node must be connected to the cluster-wide provisioning network.
- **Client Nodes** - In this installation guide, all nodes provisioned by the Cobbler/oneSIS provisioning system are referred to as clients of the head node. A client node may be provisioned in either a diskful or diskless configuration. Each client node must be connected to the cluster-wide provisioning network.

  The Cobbler provisioning tool facilitates provisioning (via DHCP/PXE) of diskful or diskless node configurations. For diskless nodes, Cobbler uses the oneSIS system administration tool to provide NFS-mounted root filesystems for each node class, such as a Lustre server, compute node, or login node.

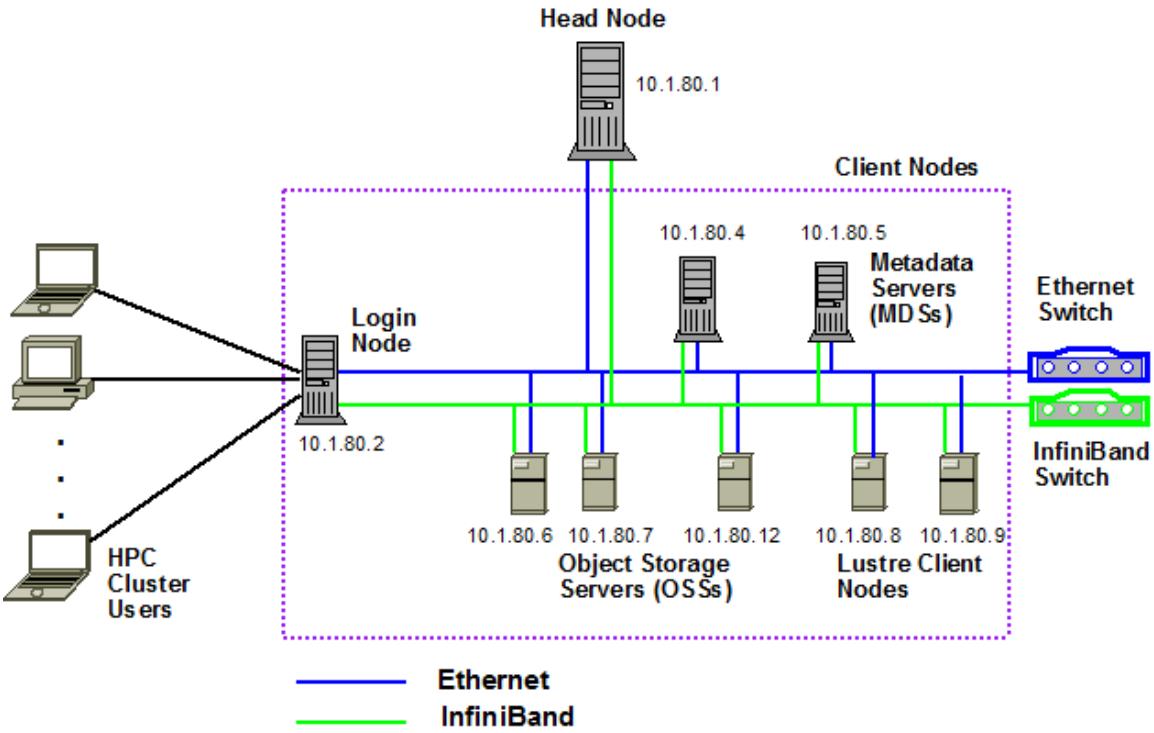

Figure 1. Example cluster configuration using an Infiniband network as the compute network

The procedure to install and configure the Sun HPC Software includes these steps:

1. Create an inventory of devices
2. Install the Sun HPC Software
3. Create configurations for all nodes in the cluster
4. Configure and start ConMan service
5. Boot the client nodes
6. Build SSH keys
7. Configure and test the operational commands
8. Perform additional site customization
9. Complete cluster-wide verification

Step 1: Create an inventory of devices

Before installing the Sun HPC Software, you will need to collect information about each node to use in subsequent steps in the procedure.

1. Create an inventory of systems in your cluster. A sample inventory is shown in Table 2.

   In this sample inventory, the node with the hostname `mgmt1` functions as the head node (see Figure 1) from which the cluster is provisioned. The nodes `login1` and `login2` correspond to the system labeled *Login Node* in Figure 1. The other nodes in Table 2 correspond to diskful (`df...`) and diskless (`dl...`) Lustre servers and their correspondent service processors. In the example cluster in Figure 1, all the nodes are connected to an Ethernet network interface and an InfiniBand network interface.

Table 1: Cluster Inventory

Management Network			InfiniBand Network	
MAC	**eth0**	**hostname**	**ib0 ip**	**ib0 name**
00:14:4f:80:14:a0	10.1.80.1	mgmt1	10.13.80.1	mgmt1-ib0
00:14:4f:82:31:5e	10.1.80.2	login1	10.13.80.2	login1-ib0
00:14:4f:9e:a0:ce	10.1.80.3	login2	10.13.80.3	login2-ib0
00:14:4f:45:26:e2	10.1.80.4	dfmds01	10.13.80.4	dfmds01-ib0
00:14:4f:45:26:e6	10.1.81.4	dfmds01-sp		
00:14:4f:11:73:45	10.1.80.5	dfmds02	10.13.80.5	dfmds02-ib0
00:14:4f:11:73:4f	10.1.81.5	dfmds02-sp		
00:14:4f:31:a0:5e	10.1.80.6	dfoss01	10.13.80.6	dfoss01-ib0
00:14:4f:31:a0:5f	10.1.81.6	dfoss01-sp		
00:14:4f:a7:30:9d	10.1.80.7	dfoss02	10.13.80.7	dfoss02-ib0
00:14:4f:a7:30:9f	10.1.81.7	dfoss02-sp		
00:14:4f:ee:6f:45	10.1.80.8	dflcn001	10.13.80.8	dflcn001-ib0
00:14:4f:ee:6f:4f	10.1.81.8	dflcn001-sp		
00:14:4f:9e:3f:f5	10.1.80.9	dflcn002	10.13.80.9	dflcn002-ib0
00:14:4f:9e:3f:fd	10.1.81.9	dflcn002-sp		
00:14:4f:45:26:d2	10.1.80.10	dlmds01	10.13.80.10	dlmds01-ib0
00:14:4f:45:26:df	10.1.81.10	dlmds01-sp		
00:14:4f:11:7e:4f	10.1.80.11	dlmds02	10.13.80.11	dlmds02-ib0
00:14:4f:11:7e:7f	10.1.81.11	dlmds02-sp		
00:14:4f:31:a0:ff	10.1.80.12	dloss01	10.13.80.12	dloss01-ib0
00:14:4f:31:a0:ef	10.1.81.12	dloss01-sp		
00:14:4f:a7:9f:9d	10.1.80.13	dloss02	10.13.80.13	dloss02-ib0
00:14:4f:a7:9f:9e	10.1.81.13	dloss02-sp		
00:14:4f:9e:6f:4f	10.1.80.14	dllcn001	10.13.80.14	dllcn001-ib0
00:14:4f:9e:6f:9f	10.1.81.14	dllcn001-sp		
00:14:4f:1e:3e:f9	10.1.80.15	dllcn002	10.13.80.15	dllcn002-ib0
00:14:4f:1e:3e:fe	10.1.81.15	dllcn002-sp		

Step 2: Install the Sun HPC Software

The Sun HPC Software installation process is designed to accommodate a variety of customer environments. Two recommended methods for installing the Sun HPC Software stack are:

- **Kickstart** – Use this method when the head node has Internet access to the Sun HPC Software repository on a Sun-hosted server. Kickstart allows a system administrator to perform a semi- or fully-automated installation of an RPM-based Linux system. A Kickstart-based installation of the Sun HPC Software results in a head node installed with the base Red Hat distribution and the Sun HPC Software stack and ready to configure.
- **DVD** – Use this method when the head node is not connected to the Internet. A DVD image (ISO) is downloaded from the Sun web site and burned to a DVD. The DVD containing the Sun HPC Software is then installed to the head node. The DVD contains all the Sun HPC software packages needed to deploy the stack on the existing Red Hat installation. It is assumed that the base Red Hat distribution is already installed on the head node.

Note: RHEL 5.2 includes OFED 1.3, which is replaced by OFED 1.3.1 when the Sun HPC Software is installed on the head node.

Installing the Sun HPC Software Using Kickstart

This procedure describes a basic Kickstart installation (see Figure 2). Prerequisites are:

- Head node has access to the public Internet
- Base RHEL 5.2 media is inserted into the head node DVD drive

Note: To ensure proper functionality of the Sun HPC Software, the head node must be registered with the Red Hat Network and all packages updated to the latest version.

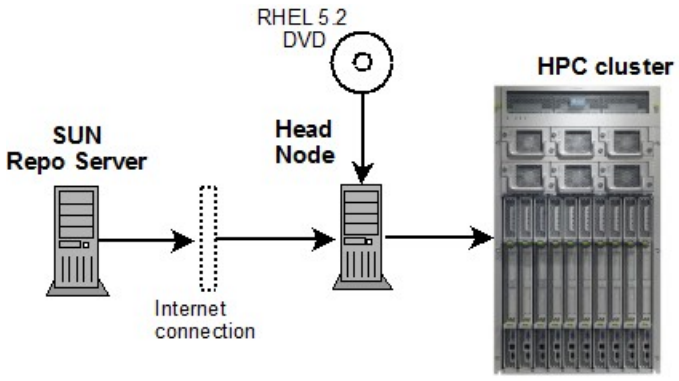

Figure 2. Installing the Sun HPC Software using Kickstart

Complete the following steps:

1. Insert the RHEL 5.2 DVD in the head node DVD drive and power on or reboot the node.

   Assuming that the BIOS has been configured to boot from the DVD device, a `boot:` prompt will appear.

2. At the `boot:` prompt,enter the following to configure the boot parameters:

```
linux ks=http://dlc.sun.com/linux_hpc/ks/rhel5.cfg ip=dhcp
```

Key parameter definitions are:

ks	Kickstart file location.
ip	IP address of the head node. If a static IP address should be used rather than a dynamically-assigned address, then substitute the correct address (for example, `10.10.10.1`) as the value of this parameter.
ksdevice	(Optional) Kickstart provisioning network device. If the default (eth0) is not correct, then set this parameter to the value of the correct network device (for example, `ksdevice=eth2`).

3. Press the <Enter> key to begin the installation process.
4. During the installation, enter site-specific values when prompted. (Other installation values will be set to optimal defaults for a Sun HPC head node.)
   - Installation number
   - Time zone
   - Disk partitions
   - Network configurations
   - `root` password

   The node will reboot.
5. Respond to any prompts for additional operating system configuration details.

   When the configuration is complete, a login prompt will appear.
6. Update the yum repository to ensure that all of the software on the system, and the kernel in particular, is up to date.

```
# yum update
```

Note: The Sun HPC Software 1.1 release supports kernel version 2.6.18-92.1.13.el5. An active Red Hat Network account is required to complete this step. If a Red Hat Network account is not available, an alternate source for base distribution packages must be added to the yum configuration.

7. Check to ensure that the new kernel was installed and set as the default boot kernel:

```
# rpm -q kernel-2.6.18-92.1.13.el5
kernel-2.6.18-92.1.13.el5

# /sbin/grubby --default-kernel
/boot/vmlinuz-2.6.18-92.1.13.el5
```

8. Reboot the head node to activate the new kernel.

9. Install Lustre Client packages for oneSIS-based clusters

```
yum -y --enablerepo=sunhpc-rhel groupinstall "SunHPC Lustre Client"
```

Continue to **STEP 3: *Create configurations for all nodes in the cluster*** to set up Cobbler on the oneSIS server and provision the cluster nodes using the Cobbler service.

Installing the Sun HPC Software from a DVD

This procedure describes how to install the Sun HPC Software from a DVD (see Figure 3). Prerequisites are:

- Head node may or may not have access to the public Internet.
- Base RHEL 5.2 distribution is already installed on the head node

Note: To ensure proper functionality of the Sun HPC Software, the head node must be registered with the Red Hat Network and all packages updated to the latest version.

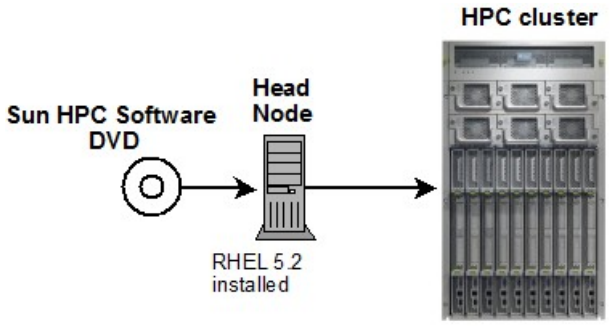

Figure 3. Installing the Sun HPC Software from a DVD

To install the Sun HPC Software from a DVD, complete the steps below.

1. Download the ISO image and burn to a DVD. The ISO image can be obtained from the Sun HPC Software, Linux Edition product page on the Sun website at http://www.sun.com/software/products/hpcsoftware/getit.jsp.

2. Insert the DVD on the head node and mount it by entering:

```
# mkdir -p /media/sun_hpc_linux
# mount -o ro /dev/dvd /media/sun_hpc_linux
# rpm -ivh /media/sun_hpc_linux/SunHPC/sunhpc-release.rpm
```

3. To install the Sun HPC Software packages that are required on the head node, start the Sun HPC Software installer by entering:

```
# sunhpc_installer
```

   The Sun HPC Software “welcome” screen is displayed as shown in Figure 4.

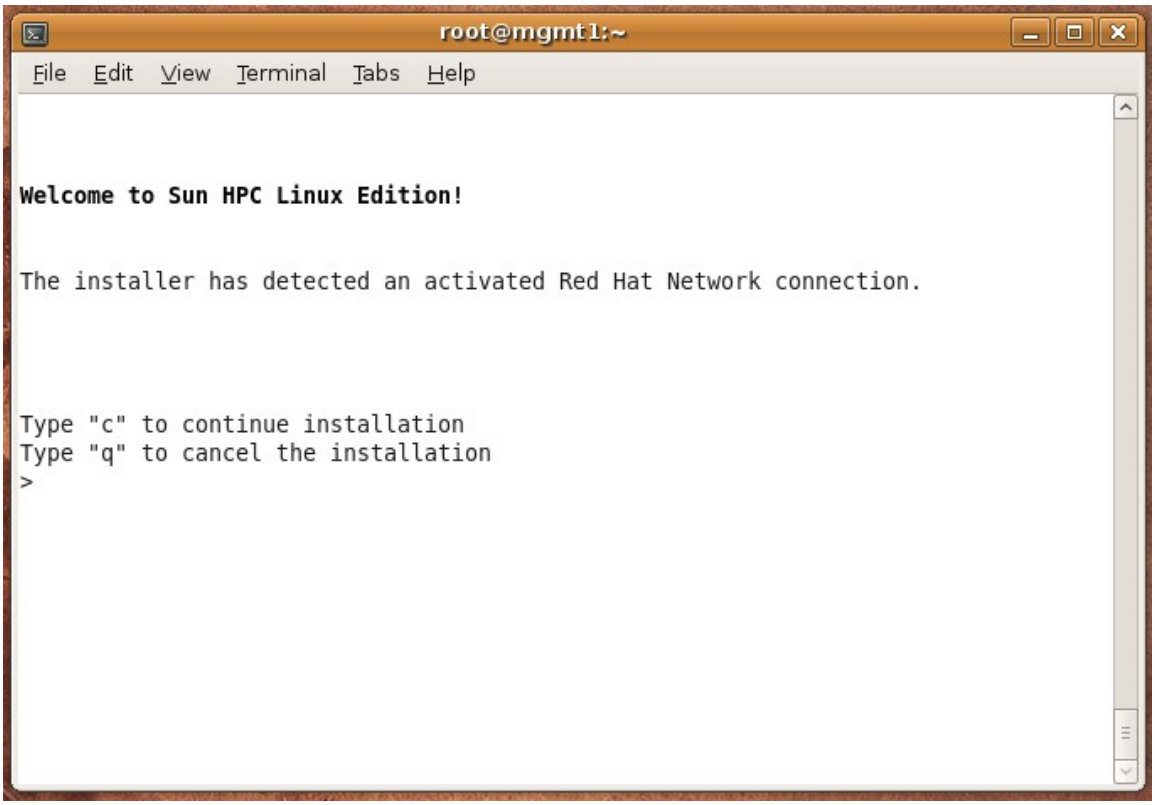

Figure 4. Sun HPC Software installer screen.

4. Enter “c” to continue. The SunHPC installer program will automatically exit if the current kernel is not supported by the Sun HPC Software 1.1 release or if a local or remote Sun HPC repository is not available.

The Sun HPC Software installer may display messages similar to the following to indicate that the installation is proceeding successfully:

- `The current kernel version 2.6.18-92.1.13.el5 is supported by the Sun HPC Software, Linux Edition 1.1.`
- `The installer has detected a valid SunHPC repository at` **`file:///media/sun_hpc_linux/`**
- **`The installer has detected an activated Red Hat Network connection.`**

- `The installer has detected that the system is running CentOS. Access to a valid CentOS 5.2 repository is required to complete this installation.`

  `Complete a "yum update" before continuing. For any questions about configuring a yum repository, please consult the CentOS documentation.`

The Sun HPC Software installer may display warning messages similar to the following to indicate a problem that will prevent successful installation of the Sun HPC Software:

- `WARNING: The installer did NOT detect an activated Red Hat Network connection. Access to a valid RHEL5.2 repository is required to complete this installation.`

  Note: Without a valid Red Hat Network subscription, the installer will be unable to satisfy dependencies from Red Hat's online package repository. It is possible to configure yum for operation with a local repository of RHEL5.2 packages, but doing so is beyond the scope of this installation guide.

- `WARNING: The current kernel version 2.6.18-92.1.13.el5 is NOT supported by the Sun HPC Software, Linux Edition 1.1. A supported kernel version is required to complete this installation.`

  `Please update your kernel to the current version. Then run sunhpc_installer again.`

Continue to **STEP 3:** ***Create configurations for all nodes in the cluster*** to set up Cobbler on the oneSIS server and provision the cluster nodes using the Cobbler service.

Step 3: Create configurations for all nodes in the cluster

Cobbler and oneSIS are used to set up the Sun HPC Software on diskful and diskless clients respectively. This step describes how to set up the Sun HPC Software using these tools.

Preparing the head node to provision diskful and diskless clients

This procedure assumes that the stack is being installed on a private, internal network. In Step 1 and 2, `iptables` are disabled to allow full access by all networking devices to the management Ethernet network.

1. Check the status of the iptables on the head node as shown below.

```
# chkconfig --list | grep tables
ip6tables       0:off   1:off   2:on    3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

2. If any `iptables` are on, disable them as shown below.

```
# /etc/init.d/iptables stop
# /etc/init.d/ip6tables stop
# chkconfig --level 2345 iptables off
# chkconfig --level 2345 ip6tables off
# chkconfig --list | grep tables
ip6tables       0:off   1:off   2:off   3:off   4:off   5:off   6:off
iptables        0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

   Security requirements for different sites vary significantly. Although configuration of `iptables` is beyond the scope of this installation guide, the following guidelines may be helpful if you need to configure an environment that requires higher security for public-facing devices, such as login nodes:

   - Use rules as appropriate to set network access for public-facing devices.
   - Run a set of commands such as the following to add rules for devices facing the private network (`eth1` in this example).

```
# iptables -A INPUT -i eth1 -j ACCEPT
# iptables -A OUTPUT -i eth1 -j ACCEPT
# /etc/init.d/iptables save
```

***Note**:* The commands shown above will completely disable the firewall on `eth1` and are intended only as an example. Please carefully consider your site's security requirements before changing `iptables` rules.

> **Cobbler** is a Linux provisioning server that provides tools for automating software installation on large numbers of Linux systems, including PXE configurations and boots, re-installation, and virtualization. For more information about using Cobbler, see https://fedorahosted.org/cobbler.

3. Modify the Cobbler settings file `/etc/cobbler/settings` as needed for your environment. Settings include:

`next_server`	IP address of the `tftp` server, which runs on the head node
`server`	IP address of Cobbler `dhcp` server, which runs on the head node
`ksdevice`	(Optional) Provisioning network interface to be used on the head node if other than the default `eth0`.

For example, to configure the head node `mgmt1` listed in Table 1, edit the `settings` file as shown here:

```
# vi /etc/cobbler/settings
-snip-
ksdevice: eth0
-snip-
next_server: '10.1.80.1'
-snip-
server: '10.1.80.1'
-snip-
```

Modify the `dhcpd.template` file for the provisioning network. In this example, the subnet starts at 10.1.0.0 and the subnet mask is 255.255.0.0. In the `dhcpd.template` file, we need to edit subnet and netmask field accordingly.

```
# vi /etc/cobbler/dhcp.template
- snip -
subnet 10.1.0.0 netmask 255.255.0.0 {
     option subnet-mask      255.255.0.0;
- snip -
```

Configuring Diskful Clients

If your system uses diskful clients, complete these steps. The `cobbler sync` command modifies the web server and `dhcp` server configurations. These servers must be restarted for the changes to take effect.

1. Restart `cobblerd`, `dhcpd`, and `httpd`.

```
# cobbler sync
# /etc/init.d/dhcpd restart
# /etc/init.d/cobblerd restart
# /etc/init.d/httpd restart
```

2. Create a new Cobbler profile for the base Red Hat distribution using one of these options:

   - Insert and mount the RHEL5.2 DVD:

```
# mount -t iso9660 -o ro /dev/cdrom /mnt/rhel5.2
# cobbler import --arch=x86_64 --name=rhel5.2 \
  --path=/mnt/rhel5.2 --kickstart=/etc/cobbler/rhel5.ks
```

   Note: Replace `/dev/cdrom` with `/dev/dvd` or `/dev/dvdw` if appropriate.

   - Mount the RHEL5.2 ISO file:

```
# mount -t iso9660  -o loop /root/ \
  RHEL5.2-Server-20080430.0-x86_64-DVD.iso /mnt/rhel5.2
# cobbler import --arch=x86_64 --name=rhel5.2 \
  --path=/mnt/rhel5.2 --kickstart=/etc/cobbler/rhel5.ks
```

3. (Optional) Create a new Cobbler repository for Red Hat updates, if present. The command line below assumes that any updated package, such as a kernel or GNU C Library (`glibc`), has been copied to the directory `/media/updates/rhel5/x86_64/RPMS`.

   Note: **Do not complete this step if updated Red Hat packages are not available.** An empty repository will cause client installations to fail.

```
# cobbler repo add --name=rhel5_updates \
  --mirror=/media/updates/rhel5/x86_64/RPMS
```

   Note: The Red Hat update repository is used for update packages provided by Red Hat that contain security and bug fixes. A support contract with Red Hat is required to download these updates.

4. Create a new Cobbler repository for Sun HPC Software.

   - *If installing from the Internet*, enter the following:

```
# cobbler repo add --name=sunhpc \
  --mirror=http://dlc.sun.com/sunhpc/yum/1.1/rhel/x86_64
# cobbler_reposync sunhpc
```

   - *If installing from a DVD*, ensure the DVD is in the drive and enter the following:

```
# mkdir -p /media/sun_hpc_linux
# mkdir -p /media/sunhpc1.1
# mount -t iso9660 -o ro /dev/dvd /media/sun_hpc_linux
# rsync -aP /media/sun_hpc_linux/ /media/sunhpc1.1/
# umount /media/sun_hpc_linux
# cobbler repo add --name=sunhpc --mirror=/media/sunhpc1.1
# cobbler_reposync sunhpc
```

   Note: The Sun custom script `/usr/sbin/cobbler_reposync` is used in place of `cobbler reposync` to create the Sun HPC Software custom package group name in the repository.

5. Modify the Cobbler profile to set client node boot parameters using one of the options below:

   - *If you are using a normal console to install your system*, enter:

```
# cobbler profile edit --name=rhel5.2-x86_64 --kopts="selinux=0" \
  --repos="rhel5_updates sunhpc"
# cobbler sync
```

   - *If you are using a serial console to install your system*, enter the following (modify the console device and speed as appropriate):

```
# cobbler profile edit --name=rhel5.2-x86_64 \
  --kopts="console=ttyS0,9600 selinux=0" \
  --repos="rhel5_updates sunhpc"
# cobbler sync
```

6. Create a Cobbler system attribute with information about the client to be provisioned, such as the client's IP address. Include the Cobbler profile, as shown below.

```
# cobbler system add --name=dflcn001 --mac=00:14:4f:ee:6f:45 \
  --ip=10.1.80.8 --subnet=255.255.0.0 \
  --hostname=dflcn001  --profile=rhel5.2-x86_64
# cobbler sync
```

   Note: By default, DHCP assigns an IP address to the provisioned operating system. To assign a permanent static IP address to the provisioned operating system, include the `--ksmeta="static=1"` option. To use `eth1`, include the `--interface=1` option.

7. Repeat Step 6 for each node to be provisioned.
8. If SELinux is enabled on the head node (run the `sestatus` command to check), the http daemon must be granted network connect privileges. To ensure that Cobbler provisioning works properly, enter the following on the head node:

```
# setsebool -P httpd_can_network_connect=1
```

9. For each client configured in step 6:
   a. Ensure that the node is set to use PXE as the default boot option.
   b. (Re-)boot the node.

   After Cobbler provisioning is complete, the client is ready for use.

   Note: New clients should be set to boot from the local disk on subsequent boots.

10. SSH may now be used to log into each node with username `root` and password `changeme`.

Continue either to the next step *(Optional) Provision Diskful Lustre Servers with Cobbler* or to *STEP 4: Configure and start Conman service*.

(Optional) Provision Diskful Lustre Servers with Cobbler

If your cluster will use the Lustre filesystem option, use the following commands to provision the Lustre MDS and OSS nodes (in addition to the nodes provisioned in the previous section).

1. Create a new system record in Cobbler for each MDS (repeat with correct values for each MDS node):

```
# cobbler system add --name=dfmds01 --mac=00:14:4f:45:26:e2 \
  --ip=10.1.80.4 --subnet=255.255.0.0 \
  --hostname=dfmds01 --profile=rhel5.2-x86_64 \
  --ksmeta="lustreserver=1 static=1"
```

2. Create a new system record in Cobbler for each OSS (repeat with correct values for each OSS node):

```
# cobbler system add --name=dloss01 --mac=00:14:4f:31:a0:ff \
  --ip=10.1.80.12 --subnet=255.255.0.0 \
  --hostname=dloss01 --profile=rhel5.2-x86_64 \
  --ksmeta="lustreserver=1 static=1"
```

3. Synchronize the changes by entering:

```
# cobbler sync
```

Configuring Diskless Clients

If your system uses *diskless clients*, complete these steps. The `cobbler sync` command modifies the web server and `dhcp` server configurations. These servers must be restarted for the changes to take effect.

1. Restart `cobblerd`, `dhcpd` and `httpd`.

```
# cobbler sync
# /etc/init.d/dhcpd restart
# /etc/init.d/cobblerd restart
# /etc/init.d/httpd restart
```

Note: For each Red Hat release, oneSIS provides a set of system patches and uses a configuration file specific to that release to create a system image. The full list of supported Red Hat releases can be found in the oneSIS installation directory on the management node (`/usr/share/oneSIS/includes/`).

> **oneSIS** is an open-source software tool for administering systems in a large-scale, Linux-based cluster environment. The default oneSIS configuration that results from building the head node is used to begin provisioning nodes in the cluster as diskless clients. More information about OneSIS can be found at http://www.onesis.org.

2. To create the oneSIS root image for diskless compute nodes, enter:

```
# onesis_setup --rootfs=/var/lib/oneSIS/image/rhel5.2 \
  --config=/usr/share/oneSIS/includes/sysimage.conf.rhel5.2 \
  --exclude=/var/www/cobbler
```

The following message will be displayed for about 10 minutes followed by several more messages:

```
Copying / to /var/lib/oneSIS/image/rhel5.2/...
```

3. Register the oneSIS root image and create a Cobbler profile for diskless clients.

   - *If you are using a normal console to install your system*, enter:

```
# cobbler distro add --name=onesis_rhel5.2 \
  --kernel=/tftpboot/vmlinuz-2.6.18-92.1.13.el5 \
  --initrd=/tftpboot/initrd-2.6.18-92.1.13.el5.img
# cobbler profile add --name=onesis_client \
  --distro=onesis_rhel5.2 \
  --kopts="selinux=0 root=/dev/nfs"
```

- *If you are using a serial console to install your system*, enter the following (modify the console device and speed as appropriate):

```
# cobbler distro add --name=onesis_rhel5.2 \
  --kernel=/tftpboot/vmlinuz-2.6.18-92.1.13.el5 \
  --initrd=/tftpboot/initrd-2.6.18-92.1.13.el5.img
# cobbler profile add --name=onesis_client \
  --distro=onesis_rhel5.2 \
  --kopts="console=ttyS0,9600 selinux=0 root=/dev/nfs"
```

4. To provision diskless clients (in this example, dlmds01, dloss01 and dllcn001), enter the following:

```
# cobbler system add --name=dlmds01 --mac=00:14:4f:45:26:d2 \
  --ip=10.1.80.10 --hostname=dlmds01 --profile=onesis_client
# cobbler system add --name=dloss01 --mac=00:14:4f:31:a0:ff \
  --ip=10.1.80.12 --hostname=dloss01 --profile=onesis_client
# cobbler system add --name=dllcn001 --mac=00:14:4f:9e:6f:4f \
  --ip=10.1.80.14 --hostname=dllcn001 --profile=onesis_client
# cobbler sync
# /etc/init.d/dhcpd restart
# /etc/init.d/cobblerd restart
# /etc/init.d/httpd restart
```

5. For each client configured in step 4:

   a. Ensure that the node has been set to use PXE as the default boot option

   b. (Re-)boot the node.

   After bootup has completed, the node will be using the NFS root filesystem and be ready for use.

6. SSH may now be used to log into each node with the username `root` and the same password that was entered during head node configuration.

Continue either to the next step *(Optional) Create Additional oneSIS rootfs Image for Lustre Servers* or to *STEP 4: Configure and start Conman service*.

(Optional) Create Additional oneSIS rootfs Image for Lustre Servers

To use the Lustre filesystem option with oneSIS as the provisioning system, create a separate oneSIS `rootfs` image for the Lustre server nodes.

1. Create a copy of the base `rootfs` and install Lustre server packages by entering:

```
# onesis_lustre_rootfs /var/lib/oneSIS/image/rhel5.2 \
  /var/lib/oneSIS/image/rhel5.2-lustre
```

2. Add a profile for diskless Lustre servers using one of these options.

   - If the server nodes have a normal console, enter:

```
# cobbler distro add --name=onesis_rhel5.2-lustre \
  --kernel=/tftpboot/vmlinuz-2.6.18-92.1.10.el5_lustre.1.6.6 \
  --initrd=/tftpboot/initrd-2.6.18-92.1.10.el5_lustre.1.6.6.img
# cobbler profile add --name=onesis_lustre_server \
  --distro=onesis_rhel5.2-lustre \
  --kopts="selinux=0 root=/dev/nfs"
```

   - If the server nodes have a serial console, enter:

```
# cobbler distro add --name=onesis_rhel5.2-lustre \
  --kernel=/tftpboot/vmlinuz-2.6.18-92.1.10.el5_lustre.1.6.6 \
  --initrd=/tftpboot/initrd-2.6.18-92.1.10.el5_lustre.1.6.6.img
# cobbler profile add --name=onesis_lustre_server \
  --distro=onesis_rhel5.2-lustre \
  --kopts="console=ttyS0,9600 selinux=0 root=/dev/nfs"
```

3. Add entries for each diskless Lustre server node.

   a. For each MDS enter the following (repeat with correct values for each MDS):

```
# cobbler system add --name=dlmds01 --mac=00:14:4f:45:26:d2 \
  --ip=10.1.80.10 –hostname=dlmds01 \
  --profile=onesis_lustre_server --dhcp-tag=lustreserver
```

   b. For each OSS enter the following (repeat with correct values for each OSS):

```
# cobbler system add --name=dloss01 --mac=00:14:4f:31:a0:ff \
  --ip=10.1.80.12 --hostname=dloss01 \
  --profile=onesis_lustre_server --dhcp-tag=lustreserver
```

4. Synchronize changes by entering:

```
# cobbler sync
# /etc/init.d/dhcpd restart
# /etc/init.d/cobblerd restart
# /etc/init.d/httpd restart
```

Step 4: Configure and start ConMan service

ConMan is a centralized console management tool capable of handling a large number of clients. It supports Sun ILOM. You can use the list of the devices with MAC and IP addresses in the spreadsheet created in STEP 1 to configure ConMan.

To configure ILOM in the ConMan serial console management tool, complete the following steps:

1. Add the nodes to the file `/etc/conman.conf` that you want to manage using ConMan.

```
# vi /etc/conman.conf

- snip -

console name="dfmds01" dev="/usr/lib/conman/exec/sun-ilom.exp \
  dfmds01-sp root"
console name="dfmds02" dev="/usr/lib/conman/exec/sun-ilom.exp \
  dfmds01-sp root"
console name="dfoss01" dev="/usr/lib/conman/exec/sun-ilom.exp \
  dfoss01-sp root"
```

2. Create a password file called `/etc/conman.pswd` with entries in the format `<host-regex> : <user> : <password>`

```
# vi /etc/conman.pswd
dfmds[01-02]*sp : root : changeme
dfoss[01-02]*sp : root : changeme
# chmod 400 /etc/conman.pswd
```

3. Start the `conman` service.

```
# /etc/init.d/conman start
```

4. Use the command `conman<hostname>` to access remote consoles being managed by the `conmand` daemon.

```
# conman node09
<ConMan> Connection to console [node09] opened.
Red Hat Enterprise Linux Server release 5.2 (Tikanga)
Kernel 2.6.18-92.1.6.el5 on an x86_64
node09 login:&.
```

Step 5: Boot the client nodes

1. Boot the client nodes in the cluster for PXE provisioning.

   The following example shows how to boot up a client for PXE provisioning. In this example, three data server nodes (identified by their ILOM IP addresses) are provisioned as diskless clients. FreeIPMI commands are used to power the nodes off and on. For more information, see the `ipmipower` man page.

```
# ipmipower -h dfmds[01-02] -sp -u root -p changeme -f -W "endianseq"
dfmds01: ok
dfmds02: ok

# ipmipower -h dlmds[01-02] -sp -u root -p changeme -n -W "endianseq"
dlmds01: ok
dlmds02: ok
```

Note: The `-W "endianseq"` option must be used for the current version of ILOM SP-based Sun Systems. It is not needed for older Sun Systems such as v20z/v40z.

Step 6: Build SSH keys

Some Sun HPC Software components, such as `pdsh`, require Secure Shell (`ssh`) public key authentication to access clients in the cluster. Follow the procedure below to set up `ssh` public keys. In the examples below, `dflcn[001-002]` are client compute nodes.

1. On the head node, create the `ssh` public key.

```
# ssh-keygen -t rsa -N ''
```

2. Copy authorized keys to the client oneSys `root` filesystem.

```
# mkdir -p /var/lib/oneSIS/image/rhel5.2/root/.ssh
# chmod 700 /var/lib/oneSIS/image/rhel5.2/root/.ssh
# cat /root/.ssh/id_rsa.pub  > \
  /var/lib/oneSIS/image/rhel5.2/root/.ssh/authorized_keys
# chmod 600 /var/lib/oneSIS/image/rhel5.2/root/.ssh/authorized_keys
```

3. Define the nodes in the cluster in `/etc/genders`.

```
# cat /etc/genders
dflcn[001-002] diskFulLustreClient
```

Note: These nodes must either be in `/etc/hosts` or must be able to be resolved through DNS.

---

4. Generate host entries defined in `/etc/genders`.

```
# ssh-keyscan -t rsa `nodeattr -s compute` > /root/.ssh/known_hosts
```

Step 7: Configure and test the operational commands

Several monitoring, provisioning, and management tools – `pdsh`, FreeIPMI, Powerman, ConMan, and Ganglia – are provided with Sun HPC Software. Follow the instructions below to configure `pdsh`, Powerman, and FreeIPMI tools as needed and test that operational commands are functional on your system. Instructions for configuring ConMan are provided in STEP 4.

Using pdsh

The `pdsh` tool allows commands to be executed at multiple nodes at the same time. `pdsh` provides a module interface and supports `rsh` and `ssh` for communication. In the examples below, `dfmds[01-02]` are metadata server nodes. A command line example is shown below:

```
# pdsh -w dfmds[01-02] uptime
dfmds01:  21:10:46 up  5:28,  0 users,  load average: 0.00, 0.00, 0.00
dfmds02:  21:11:49 up  5:29,  0 users,  load average: 0.00, 0.00, 0.00
```

To use a `dsh` (Dancer's shell) group file with `pdsh`, create an `/etc/dsh/group` directory. For each group, place a file in this directory that contains the hostname of each node in the group. In the example below, three groups are created named “client”, “oss”, and “mds”:

```
# mkdir -p /etc/dsh/group
# vi client
dflcn001
dflcn002

# vi oss
dfoss01
dfoss02

# vi mds
dfmds01
dfmds02
```

In the example below, `pdsh` is used with the `-g groupname` option:

```
# pdsh -g all uptime
dfmds01:  21:17:25 up  5:34,  0 users,  load average: 0.03, 0.01, 0.00
dfmds02:  01:14:29 up  5:34,  0 users,  load average: 0.08, 0.02, 0.01
dfoss01:  00:16:25 up  5:34,  0 users,  load average: 0.04, 0.01, 0.00
dfoss02:  21:16:23 up  5:34,  0 users,  load average: 0.00, 0.00, 0.00
dflcn001:  21:15:33 up  5:34,  0 users,  load average: 0.00, 0.00, 0.00
dflcn002:  21:17:13 up  5:34,  0 users,  load average: 0.00, 0.00, 0.00

# pdsh -g oss uptime
dfoss01:  21:16:58 up  5:35,  0 users,  load average: 0.00, 0.00, 0.00
dfoss02:  00:17:01 up  5:35,  0 users,  load average: 0.02, 0.01, 0.00
```

Using PowerMan

PowerMan is used to operate remote power control (RPC) devices from a central location. You can use the list of the devices with MAC and IP addresses in the spreadsheet created in STEP 1 to configure PowerMan. In the examples below, nodes `dfmds[01-02]` are diskful metadata server nodes.

Configuring the PowerMan power management tool

a. Edit the PowerMan configuration file.

```
# vi /etc/powerman/powerman.conf
include "/etc/powerman/ipmipower.dev"
alias "dfmds" "dfmds[01-02]"
device "pow0" "ipmipower"  "/usr/sbin/ipmipower -h dfmds[01-02]-sp |&"
node "dfmds[01-02]" "pow0"  "dfmds[01-02]-sp"
```

b. Start Powerman.

```
# /etc/init.d/powerman start
```

Displaying the current power status of the `dfmds`) nodes

```
# powerman -q dfmds
on:      [dfmds01-dfmds02]
off:
unknown:
```

Shutting down all `dfmds` nodes and displaying the power status again

```
# pdsh -w dfmds[01-02] shutdown -h now
# powerman -q all
on:
off:     [dfmds01-dfmds02]
unknown:
```

Turning on power for all `dfmds` nodes

```
# pm -1 dfmds
Command completed successfully
```

Note: The `pm` command is an alias for the `powerman` command.

Using FreeIPMI

FreeIPMI provides support for Sun's ILOM with the `-u root` option and the `-W "endianseq"` option. You can enter these as command line options or incorporate them into a configuration file. A command line example is shown below.

```
# ipmipower -h 10.1.81.[4-5]  -p changeme -u root -s -W "endianseq"
10.1.81.4: on
10.1.81.5: on

# ipmi-chassis -h 10.1.81.[4-5]  -p changeme -u root -s -W "endianseq"
10.1.81.4: System Power               : on
10.1.81.4: System Power Overload      : false
10.1.81.4: Interlock switch           : Inactive
10.1.81.4: Power fault detected       : false
10.1.81.4: Power control fault        : false
10.1.81.4: Power restore policy       : Unknown
10.1.81.4: Last Power Event           : power on via ipmi command
10.1.81.4: Misc Chassis status        :
10.1.81.4: Front panel capabilities   :
10.1.81.5: System Power               : on
10.1.81.5: System Power Overload      : false
10.1.81.5: Interlock switch           : Inactive
10.1.81.5: Power fault detected       : false
10.1.81.5: Power control fault        : false
10.1.81.5: Power restore policy       : Unknown
10.1.81.5: Last Power Event           : power on via ipmi command
10.1.81.5: Misc Chassis status        :
10.1.81.5: Front panel capabilities   :
```

The example belows shows how ILOM options are defined in the configuration file `/etc/ipmipower.conf`.

```
hostname 10.1.81.[4-5]       # client's hostname
username root                   # ILOM's username
password changeme               # ILOM's password
workaround-flags "endianseq"    # work-around flags for ILOM
on-if-off enable
wait-until-on enable
wait-until-off enable
```

Once you've set up the configuration file as described above, you can use commands similar to the example below, which uses the `-s` option get the power status of the hosts specified in the configuration file.

```
# ipmipower -s
10.1.81.4: on
10.1.81.5: on
```

Using Ganglia

Ganglia is a monitoring tool for clusters. In the example below, the head node is set up to use the Ethernet 1 interface for ganglia monitoring. In the examples below, nodes `dfmds01` and `dfoss[01-02]` are metadata server and object storage server nodes.

```
# pdsh -w dfmds01,dfoss[01-02] route add -net 224.0.0.0 netmask 240.0.0.0 \
  dev eth1
# echo "any net 244.0.0.0 netmask 240.0.0.0 dev eth1" > \
  /var/lib/oneSIS/image/rhel5.2/etc/sysconfig/static-routes
```

You can use an advanced ganglia configuration for your cluster environment, but the simplest configuration assumes a single cluster name. This cluster name is defined by modifying `/etc/gmond.conf` on each client.

```
# vi /var/lib/oneSIS/image/rhel5.2/etc/gmond.conf
- snip -

 * NOT be wrapped inside of a <CLUSTER> tag. */
cluster {
  name = "giraffe_cluster"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}
- snip -
```

Use `pdsh` to start the Ganglia daemon `gmond` on all nodes in the cluster.

```
# pdsh -w dfoss[01-02] /etc/init.d/gmond start
```

On the head node, in `/etc/gmetad.conf`, add the client name to `data_source` and change the `gridname` to the name of the cluster.

```
# vi /etc/gmetad.conf
- snip -

# data_source "my cluster" 10 localhost  my.machine.edu:8649  1.2.3.5:8655
# data_source "my grid" 50 1.3.4.7:8655 grid.org:8651 grid-backup.org:8651
# data_source "another source" 1.3.4.7:8655  1.3.4.8

data_source "dfoss01" dfoss01
data_source "dfoss02" dfoss02
data_source "dflcn001" dflcn001
data_source "dflcn001" dflcn002
- snip -

# The name of this Grid. All the data sources above will be wrapped
in a GRID
# tag with this name.
# default: Unspecified
gridname "giraffe_cluster"
#
- snip -
```

Start the Ganglia daemon `gmetad` on the head node.

```
# /etc/init.d/gmetad start
```

Step 8: Perform additional site customization

Complete additional site customization such as configuring an InfiniBand network or implementing and configuring a scheduler.

Note: The Sun Grid Engine scheduler can be used with the Sun HPC Software. For more information about Sun Grid Engine, visit http://www.sun.com/software/gridware/. User documentation can be found at http://docs.sun.com/app/docs/coll/1017.4?l=en.

Step 9: Complete cluster-wide verification

Complete verification procedures as needed at your site to determine if the Sun HPC Software and third-party components have been installed and are inter-operating correctly.

Appendix A: Sun HPC Software Component Descriptions

Conman 0.2.1 – ConMan is a serial console management program designed to support a large number of console devices and simultaneous users. (http://home.gna.org/conman/)

env-switcher 1.0.13 – Environment Switcher is a thin layer on top of the modules package that allows users to manipulate the environment that is loaded for all shells (including non-interactive remote shells) without manually editing their startup dot files. (http://sourceforge.net/projects/env-switcher/)

FreeIPMI 0.6.6 – FreeIPMI is a collection of Intelligent Platform Management Interface (IPMI) system software that provides in-band and out-of-band software and a development library conforming to the Intelligent Platform Management Interface (IPMI v1.5 and v2.0) standards. (http://www.gnu.org/software/freeipmi/)

Ganglia 3.0.7 – Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and grids. (http://ganglia.info/)

Genders 1.9 – Genders is a static cluster configuration database used for cluster configuration management. (https://computing.llnl.gov/linux/genders.html)

Heartbeat 2.1.3 – Heartbeat is a GPL-licensed portable cluster management program for high-availability clustering. (http://www.linux-ha.org/Heartbeat)

HPCC 1.2.0 – HPC Challenge is a collection of benchmarks for measuring various aspects of system performance, such as flop/s, sustainable memory bandwidth, memory read/write rates, network bandwidth, and latency for parallel machines. (http://icl.cs.utk.edu/hpcc/)

IOR 2.1 – Interleaved-Or-Random Filesystem Benchmarking software is used for benchmarking parallel file systems using POSIX, MPIIO, or HDF5 interfaces. (http://sourceforge.net/projects/ior-sio)

lshw B.02.12.01 – lshw (Hardware Lister) provides detailed information on the hardware configuration of a machine, such as exact memory configuration, firmware version, mainboard configuration, CPU version and speed, cache configuration, and bus speed, on DMI-capable x86 or EFI (IA-64) systems and on some PowerPC machines. (http://ezix.org/project/wiki/HardwareLiSter)

Lustre 1.6.6 – Lustre is a scalable, secure, robust, highly-available cluster file system designed, developed and maintained by Sun Microsystems, Inc. (http://wiki.lustre.org/index.php?title=Main_Page)

Modules 3.2.6 – The Environment Modules package provides for the dynamic modification of a user's environment using module files. (http://modules.sourceforge.net/)

MUNGE 0.5.8 – MUNGE (MUNGE Uid 'N' Gid Emporium) is an authentication service for creating and validating credentials designed to be highly scalable for use in an HPC cluster environment. (http://home.gna.org/munge/)

MVAPICH 1.0.1 – MVAPICH is a MPI-1 implementation based on MPICH and MVICH that supports a variety of transport interfaces on a wide range of platforms. The name comes from the abbreviation of MPI-1 over OpenFabrics/Gen2, OprnFabrics/Gen2-UD, uDAPL, InfiniPath, VAPI and TCP/IP. (http://mvapich.cse.ohio-state.edu/index.shtml)

MVAPICH2 1.0.3 – MVAPICH2 is an MPI-2 implementation based on MPICH2 and MVICH. It backward supports all MPI-1 features. It supports several transport interfaces including OpenFabrics-IB, OpenFabrics-iWARP, uDAPL, and TCP/IP. (http://mvapich.cse.ohio-state.edu/index.shtml)

NetPIPE 3.7.1 – Network Protocol Independent Performance Evaluator (NetPIPE) is a protocol independent performance tool that visually represents the network performance under a variety of conditions. (http://www.scl.ameslab.gov/netpipe/)

OFED 1.3.1 – The OpenFabrics Enterprise Distribution (OFED) is a validated version of the open-source OpenFabrics software stack that supports server and storage clustering and grid connectivity using RDMA-based InfiniBand and iWARP fabrics in a Linux environment. (http://www.openfabrics.org)

OneSIS 2 – OneSIS is an open-source software package aimed at simplifying diskless cluster management. (http://www.onesis.org)

Open MPI 1.2.6 – The Open MPI Project is an open source MPI-2 implementation developed and maintained by a consortium of academic, research, and industry partners. (http://www.open-mpi.org)

pdsh 2.14 – Parallel Distributed Shell (pdsh) is an efficient, multi-threaded remote shell client that executes commands on multiple remote hosts in parallel. pdsh implements dynamically loadable modules for extended functionality such as new remote shell services and remote host selection. (http://sourceforge.net/projects/pdsh/)

Powerman 1.0.32 – PowerMan is a tool for manipulating remote power control (RPC) devices from a central location. (http://powerman.sourceforge.net/)

RRDtool 1.2.26 – Round Robin Database tool stores and retrieves data from Round Robin Databases (RRDs). (http://oss.oetiker.ch/rrdtool/)

SLURM 1.3.6 – The Simple Linux Utility for Resource Management (SLURM) is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. (https://computing.llnl.gov/linux/slurm/)

Appendix B: Glossary

Cfengine 2.2.6 – Cfengine is an automated suite of programs for configuring and maintaining Unix-like computers. (http://www.cfengine.org)

Cobbler is a Linux provisioning server that provides tools for automating software installation on large numbers of Linux systems, including PXE configurations and boots, re-installation, and virtualization. For more information about using Cobbler, see https://fedorahosted.org/cobbler.

Dynamic Host Configuration Protocol (DHCP) is used by clients to obtain the parameters needed to operate in an Internet Protocol (IP) network, allowing devices to be added to the network with minimal or no manual configuration.

Kickstart is a method for automating the installation of Red Hat Enterprise Linux. A configuration file is used that contains the answers to questions that would normally be answered interactively during a typical installation.

oneSIS is an open-source software tool for administering systems in a large-scale, Linux-based cluster environment. The default oneSIS configuration that results from building the head node is used to begin provisioning nodes in the cluster as diskless clients. More information about OneSIS can be found at http://www.onesis.org.

Preboot eXecution Environment (PXE) is a client/server interface that allows networked computers without an operating system installed to be configured and booted remotely.

Red Hat Package Manager (RPM) refers to a software package file format used primarily for Linux distributions and also to the software packaged in this format.

Yellow Dog Updater, Modified (YUM) is an open source, command line, package management utility for RPM-compatible Linux operating systems.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program,

and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or

distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" 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. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO

MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA *
Phone: 1-650-960-1300 or 1-800-555-9SUN (9786) Web: sun.com