NIFLHEIM

SystemImager software

The NIFLHEIM cluster uses the SystemImager toolkit on a central server to create an image of a Golden Client node that has been installed in the usual way using a distribution on CD-ROM (Redhat 7.3 in our case). The SystemImager is subsequently used to install identical images of the Golden Client on all of the nodes (changing of course hostname and network parameters). Sometimes you may want to download Beta versions or utilities. There is a useful manual for the SystemImager software which you are advised to study and use for your cluster installation.

Based on our experiences with SystemImager, we offer some additional advice that may be of use to other cluster builders:

Populating hosts files, DNS etc.

When you have a large number of nodes in your cluster, it becomes a major burden just typing in the following files files a) /etc/hosts, b) /etc/hosts.equiv, c) the PBS-server nodes-file, d) the SSH ssh_known_hosts file, e) the DNS zone-file and inverse zone-files, and f) a list of adhesive labels for the nodes. The SystemImager addclients tool will actually populate the /etc/hosts file, but for the remaining tasks we have written a simple C-code clusterlabel.c which generates this information. Just edit this code to use names appropriate for your cluster, and then run the code a number of times with different flags in order to generate your configuration files.

RPM software packages

Download the additional RPM packages specified by the SystemImager download page, which typically refers to downloads from http://rpmfind.net/. Especially important is that you get a recent version of rsync, both because of security holes that were fixed in rsync, and

because of robustness.

Node numbering

The SystemImager addclients script insists that nodes should be numbered without leading zeroes (e.g., n1, n2, ..., n99). I disagree with this policy and prefer a homogeneous numbering with the same number of digits on all nodes (e.g., n001, n002, ..., n480). If you wish, you can fix this in the addclients script around line 365 to read in stead:

```
$starting_number = get_response($starting_number);
print "What number should I end with? [$ending_number]: ";
$ending_number=get_response($ending_number);
```

Test installation

Install the SystemImager packages on the server and Golden Client as described in the documentation. Make sure that the Golden Client's network configuration has been set up for assigning IP address information by means of DHCP. Extract an image of the Golden Client to the server using the getimage utility. Now you're ready to install additional clients.

First of all, we suggest that you try to install new clients using the SystemImager boot diskette, as created by the mkautoinstalldiskette tool. This is the simplest way to test that cloning of the Golden Client actually works correctly. When this process works smoothly, you may consider network booting of nodes as described later.

Diskette booting of nodes

For smaller clusters it may make sense to let the PC boot using a special floppy diskette with a minimal Linux that will transfer a Linux Golden Client image from the server. The challenge here may possibly be to have a Linux kernel that supports your hardware, yet be compact enough to fit on a 1.44 MB diskette. With a bit of luck, the default boot diskette created by mkautoinstalldiskette will support your hardware correctly. However, a lot of the discussion on the SystemImager mailing list has been concerned with kernel support of various Ethernet cards.

We use the following procedure for creating a SystemImager boot diskette with the latest Linux kernel. This has enabled us to create a diskette that will boot and install a Compaq EVO D510, which needs kernel 2.4.19 or newer for automatic recognition of the Ethernet interface in the i845G chipset:

1. Download a 2.4.19 kernel from http://www.kernel.org/ or a mirror such as ftp.funet.fi.

2. Unpack the kernel to $KERNELDIR (let's say /tmp/linux-2.4.19)

3. Copy the SystemImager kernel config file (we provide a sample config file) to the file $KERNELDIR/.config (notice the name dot-config).

4. cd $KERNELDIR

5. make oldconfig
   Here you may decide to change the kernel configuration for other types of hardware by running in stead:
   make menuconfig

6. make dep

7. make bzImage

8. cp /usr/local/share/systemimager/i386-boot/kernel /usr/local/share/systemimager/i386-boot/kernel.orig

9. cp $KERNELDIR/arch/i386/boot/bzImage /usr/local/share/systemimager/i386-boot/kernel
   For network booting you need also to make this copy:
   cp $KERNELDIR/arch/i386/boot/bzImage /tftpboot/kernel_2.4.19.

At this point you should have a working kernel. Now you need to fix the /usr/local/share/systemimager/i386-boot/initrd.gz ram-disk file to work with the newer kernel:

1. cp -p /usr/local/share/systemimager/i386-boot/initrd.gz /usr/local/share/systemimager/i386-boot/initrd.gz.orig

2. cp /usr/local/share/systemimager/i386-boot/initrd.gz /tmp

3. gunzip /tmp/initrd.gz

4. mount /tmp/initrd /mnt -o loop

5. Edit /mnt/etc/init.d/rcS. In this file the command "mount /dev/ram1 /tmp" should be changed into "mount /dev/ram1 -t ext2 /tmp"

6. Save the file and exit the editor.

7. umount /mnt

8. gzip /tmp/initrd

9. cp /tmp/initrd.gz /usr/local/share/systemimager/i386-boot/initrd.gz
   For network booting you need also to make this copy:
   cp /tmp/initrd.gz /tftpboot/initrd.gz

Now the initrd.gz ram-disk is working with the kernel version 2.4.X. You only need to use mkautoinstalldiskette to create a working boot floppy. Boot a clean machine from the floppy and watch it install ! Any problems with this procedure should be reported to the SystemImager mailing-list.

Network booting of nodes

SystemImager allows you to boot and install nodes using the nodes' Ethernet network interface. You will be using PXE, the Intel-defined Pre-Boot eXecution Environment which is implemented in many Ethernet chips. The following advice works correctly for recent PCs, such as the Compaq EVO D510 whose Ethernet chip implements the PXE Boot Agent version 4.0.22. For older version of PXE you may have to install a pxe daemon RPM on the server (not discussed any further here), but the pxe daemon is not necessary with newer PXE versions.

Booting Linux for network installation uses the SYSLINUX and PXELINUX utilities. Read the documentation on these pages to get an understanding of the process.

Modifications of Redhat daemons

For installation with a Redhat SystemImager server, the syslinux-1.52 RPM that comes with Redhat 7.3 is no good, as we found out the hard

way. Go to rpmfind.net and download and install the latest syslinux for Redhat (rawhide); currently that is syslinux-1.75.

Copy the file /usr/lib/syslinux/pxelinux.0 to the /tftpboot directory for network booting. Also, from the SystemImager boot diskette created previously, copy the file message.txt to the /tftpboot directory. In the syslinux directory /tftpboot/pxelinux.cfg/ you should create the syslinux configuration file named default containing these lines:

```
default netboot
label netboot
kernel kernel_2.4.19
append vga=extended load_ramdisk=1 prompt_ramdisk=0 initrd=initrd.gz
root=/dev/ram rw
DISPLAY message.txt
PROMPT 1
TIMEOUT 50
```

This file as well as message.txt, the kernel and initrd.gz will be downloaded by the client node using TFTP as part of the PXE booting process.

Also, you have to replace the Redhat 7.3 TFTP server by a better one that supports the "tsize" TFTP option (RFC 1784/RFC 2349). The SYSLINUX documentation suggests several different TFTP servers, among others the atftp by Jean-Pierre Lefebvre, which we have decided to use. We compile the TFTP-server and install it as /usr/local/sbin/in.tftpd, then modify the /etc/xinetd.d/tftp configuration file like this:

```
server = /usr/local/sbin/in.tftpd
server_args = --maxthread 1000 --no-multicast
```

and restart the xinetd daemon.

DHCP setup

The DHCP server daemon should be configured correctly for booting and installation of client nodes, please see the syslinux documentation. Our Redhat 7.3 has the dhcp-2.0 RPM installed and the following configuration in /etc/dhcpd.conf:

```
# make network booting the SystemImager autoinstallclient possible
allow booting;
```

```
allow bootp;
# set lease time to 3 days
default-lease-time 259200;
max-lease-time 259200;
subnet 10.1.0.0 netmask 255.255.0.0 {
deny unknown-clients;
option domain-name "dcsc.fysik.dtu.dk";
option domain-name-servers 10.1.128.2;

option time-offset 1; # Middle European Time (MET)
option ntp-servers 10.1.128.2; # ntp.darenet.dk
group {
# option-100 specifies the IP address of your SystemImager image server
option option-100 "10.1.128.2";
next-server 10.1.128.2;
filename "/pxelinux.0";
host n001 { hardware ethernet 00:08:02:8e:05:f2; fixed-address
n001.dcsc.fysik.dtu.dk;}
# Lots of additional hosts...
}
}
```

Of course, you have to change IP addresses and domain-names for your own cluster. The client nodes' Ethernet MAC-addresses must be configured into the /etc/dhcpd.conf file. Alternatively, you can let the DHCP server hand out IP addresses freely, but then you may loose the ability to identify nodes physically from their IP addresses. We recommend to use the statically assigned IP addresses in the /etc/dhcpd.conf. This can be achieved by the following procedure:

1. Configure the DHCP server without the clients' MAC-addresses and use the deny unknown-clients option in the configuration file.

2. Connect the client nodes to the network and turn them on one by one. In the NIFLHEIM installation we did this as part of the setup process, at the same time as we customized the BIOS settings.

3. For all the client node names, note in the server's /var/log/messages file the client's Ethernet MAC-address. Label each client node with an adhesive label containing the correct node name.

4. In a file with a list of client node names you add the MAC-address to the node's line in the file.

5. When all nodes have been registered, use a simple awk-script or similar to convert this list into lines for the /etc/dhcpd.conf file, such as this one:
   host n001 { hardware ethernet 00:08:02:8e:05:f2; fixed-address n001.dcsc.fysik.dtu.dk;}

If your cluster is on a private Internet (such as the 10.x.y.z net) and your DHCP server has multiple network interfaces, you must make sure that your DHCP server doesn't offer DHCP-service to the non-cluster networks (a sure way to find a lot of angry colleagues before long :-). Edit the Redhat 7.3 configuration file /etc/sysconfig/dhcpd to contain:

DHCPDARGS=eth1

(where eth1 is the interface connected to your cluster) and restart the dhcpd daemon.

Network installation of nodes

With the above setup you're now ready to boot and install a fresh node across the network. Make sure that the PC BIOS has been set up for a boot order where network/PXE boot precedes booting from hard disk. Use a screen to monitor the installation process.

The installation process looks just like the one you have tested using the boot diskette method, except that the Ethernet adapter will now request and receive DHCP network configuration information from your server. Monitor the server's /var/log/messages file to ensure that the client node requests and is assigned a proper IP address. The client node's PXE firmware will now transfer the small Linux kernel and ram-disk and begin the installation process by transferring the Golden Client image. When you see that the node is ready to be rebooted, you do a power cycle and go into BIOS setup mode. Here you must change the boot order so that network/PXE booting no longer preceeds the booting from hard disk. Reboot the node, and watch it boot Linux from its own hard disk. The IP address should be assigned correctly by the DHCP server.

Automated network installation.

Having to watch the installation process and finally change the client nodes' BIOS setup is cumbersome when you have more than a dozen or

two client nodes. After having tested the network installation process manually as described above, you can automate the process completely using the pxeconfig toolkit written by Bas van der Vlies. Now a client node installation is as simple as configuring on the central server whether a node should perform a network installation or simply boot from hard disk: When the node is turned on, it all happens automatically with no operator intervention at all !

Download the pxeconfig toolkit and read the INSTALL instructions. Rename the above mentioned file /tftpboot/pxelinux.cfg/default as /tftpboot/pxelinux.cfg/default.node_install.

Use pxeconfig to configure those client nodes that you wish to install (the remaining nodes will simply boot from their hard disk). The pxeconfig tool creates soft-links in the /tftpboot/pxelinux.cfg directory named as the hexadecimally encoded IP-address of the clients, and these links will point to the file default.node_install. As designed, the PXE network booting process will download the file given by the hexadecimal IP-address, and hence network installation of the node will take place.

The second part of the pxeconfig toolkit is the pxeconfigd daemon, which will remove the hexadecimally encoded IP-address soft-link on the server when contacted by the client node. In order for this to happen, you must go to the server's /var/lib/systemimager/scripts directory and edit the image-specific script <imagename>.master. Near the end of this script you will see the lines:

```
# Take network interface down
ifconfig eth0 down || shellout
```

Just before these lines insert the following lines:

```
echo "Contacting server for pxeconfig and then reboot"
telnet $IMAGESERVER 10000
sleep 1
reboot
```

The telnet connection to the server will make the pxeconfigd daemon remove the soft-link for this particular IP address. When the reboot follows, the client node's PXE booting will now download the default file, which is a soft-link to the file default.harddisk from the pxeconfig toolkit. The default.harddisk file:

```
default harddisk
label harddisk
```

localboot 0

instructs the node to boot from its local disk as described in the syslinux documentation.

Advice for large clusters

The above procedure has worked beautifully for installing our 480-node Linux cluster. However, you should divide node installation into smaller groups of nodes, since the image server can only serve node images to a limited number of clients at a time owing to the server's finite network and CPU capacity. If you install N nodes simultaneously, you will load the server's network by up to N times 100 Mbit/sec (assuming client nodes with Fast Ethernet), and the server's CPU load of N running rsync daemons will be approximately N. Therefore you should have a fairly powerful server, unless you decide to install the client nodes one by one.

In the case of the NIFLHEIM cluster, our dual-processor Pentium Xeon 2.4 Ghz server with a Gigabit network interface performed reliably when we installed 18 client nodes simultaneously. Owing to the structure of the client nodes' electrical power supply, we can turn groups of 18 nodes on and off by a single power switch. The time to install 18 clients each with a 1.5 GB disk image is about 6 minutes in our configuration.

Center for Atomic-scale Materials Physics homepage