Redbooks Paper

Bob Maher

Techniques to Build a Diskless Boot Linux Cluster of JS21 Blades

Blades are becoming an increasingly popular method for companies to use to build up their infrastructure at a low cost. Blades provide a large computing powerhouse in an extremely small, condensed footprint. Using Linux® as the operating system of choice and booting "diskless" provides for even lower cost of ownership, faster build times, and easier maintenance cycles.

This Redpaper describes the experience of configuring the IBM® Diskless Image Management (DIM) Server and a cluster of System p-based JS21 blades to remotely boot from the DIM Server (diskless boot). The paper gives examples that explain and demonstrate how you can build a similar cluster.

Introduction

When running on a diskless boot infrastructure, the master node becomes the key place to install application code and apply updates. All updates are done on a single node and the data is replicated to each participating blade in the complex, thus putting an end to the task of ensuring that all nodes in the complex have the same levels of software. In addition, if a blade happens to go down because of a hardware issue, new hardware can be swapped out with no impact to the operating system (OS), application code, or data. Since the booting is remote, the blade is back online as soon as the new hardware is replaced and the blade is booted.

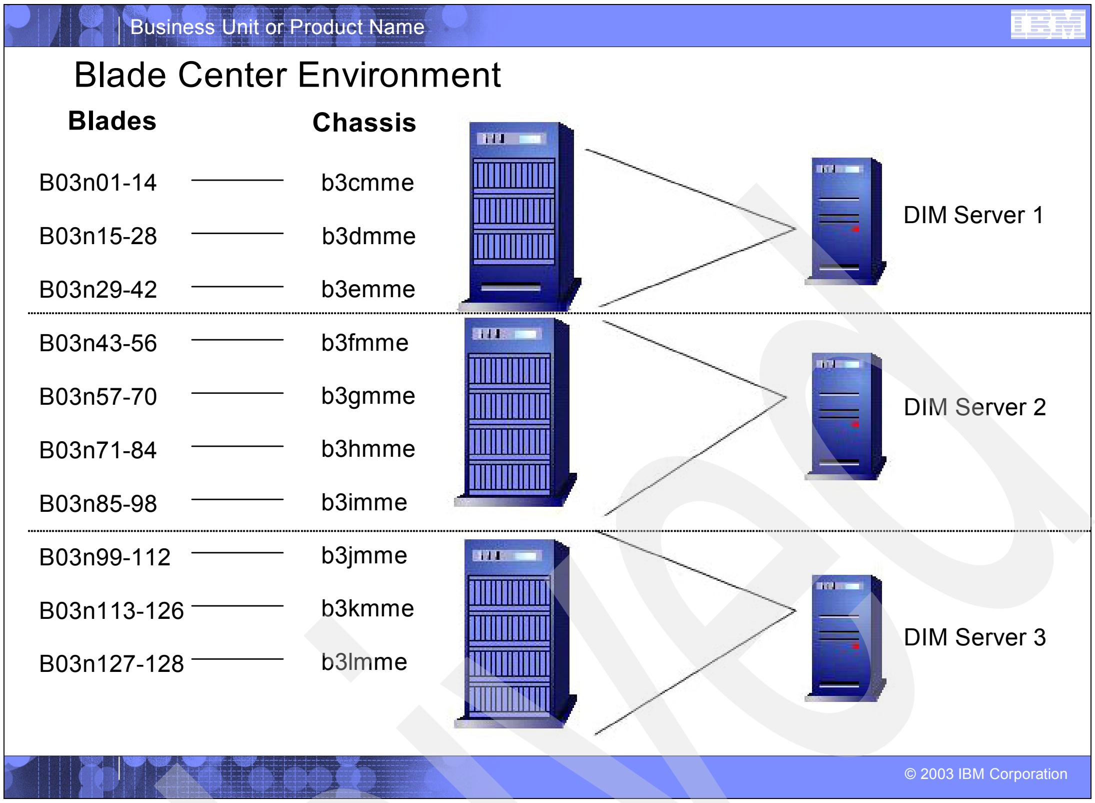
The objective of this Redpaper is to describe in detail the steps we used to install and configure our IBM BladeCenter® complex, which remotely boots the blades using the SUSE SLES9 Operating System from the Diskless Image Management (DIM) Server. Our blade environment for testing consisted of 128 JS21 System p blades in a BladeCenter H chassis. The BladeCenter H chassis is the new chassis designed for the increased cooling and power requirements for the JS21 blades.

The DIM Server provides the means to boot the blades using DHCP served from the DIM Server. The blade is booted from an image file and the kernel is loaded into memory while the rest of the required files (such as /usr, /etc, /opt, and so on) are mounted from the DIM Server. The DIM Server also provides tools that simplify the ability to remotely query the BladeCenter chassis to perform tasks such as find blade status information, temperature, network address, and more.

A mini "how-to" document about this topic is provided by the DIM Server developer, Peter Morjan. This Redpaper was written to provide more detail on the specifics of our build in support of our high performance computing environment and illustrate how you can build a similar cluster.

Topology

Figure 1 on page 3 shows the environment we set up. We grouped the servers into three groups based on the DIM Server to which they were attached. It just happened that the blades attached to each DIM Server were each on the same rack. There are 14 blades per BladeCenter Chassis. This made the management that much easier.



Business Unit or Product Name

Blade Center Environment

Blades	Chassis
B03n01-14	b3cmme
B03n15-28	b3dmme
B03n29-42	b3emme
B03n43-56	b3fmme
B03n57-70	b3gmme
B03n71-84	b3hmme
B03n85-98	b3imme
B03n99-112	b3jmme
B03n113-126	b3kmme
B03n127-128	b3lmme

Figure 1 Our blade environment

We planned for a maximum of 56 blades to be attached to any one DIM server. This ensures that performance will not be affected by having too may nodes connected to the nfs mount points on the DIM Server. Additionally, this provides a smaller impact to the number of blades that will be out of service at any given point, should the DIM Server experience problems.

In this environment, each blade queries the DIM Server to obtain a TCP/IP address. After the DHCP server on the DIM Server assigns the address, the blade mounts the nfs share from the DIM Server. The DIM Server dhcpd.conf file contains the information to point the blade to the appropriate mount point from which to boot the Linux image.

Booting blades from the remote boot server provides a very fast way to reach a running state from the start. Because the image and all the software, drivers, and supporting tools are installed on the master blade, the install of the master is the most time-consuming part of the install process. This is a one-time install that is done as carefully as possible.

After the install and customization, the image is created and the boot image is pushed out to the boot servers. The build of the boot servers is somewhat more complicated, but there are many tools to help this task go smoothly. After the infrastructure is in place, the build of each blade added to the system is as simple as adding the blade to the chassis and powering on. The image is transferred at boot time and the blade boots exactly as the master was built.

Maintenance of the environment is usually done from the master blade. Any new software or drivers is loaded on the master blade and using the syncing utilities of the boot server, each individual blade can be synced with the image. There are sync utilities included with the boot server to synchronize the images with each of the multiple boot servers, as well. In this way, you only need to log onto one of the boot server machines to issue the commands to synchronize the environment.

The following section describes the steps needed to create and maintain the environment.

Prerequisite activities

The following activities were completed in order to proceed with the build process:

- The BladeCenter tower, chassis, and power were put on the floor and the network cabling was provided.
- The DIM Servers were also put in place and a network was provided for each interface that the blades used.
  - A separate network address was provided for each network interface card in the blade chassis, including the Myrinet adapters
  - In our case we did not have a Myrinet adapter for the DIM Server, so the addressing for the adapter on the blades was provided in the file /etc/boot.local.
- The required software was installed on the DIM Server boxes.

  We installed dim-0.9.6-1.
- A “master blade” was installed and prepared with all the required software for our environment.

  We used the first blade in the first chassis of our environment as the master blade.

DIM server build procedures

In the following section we describe how we built our servers.

DIM terminology conventions

DIM Server The server that provides NFS, TFTP, and DHCP.
DIM compute node The system (blade) that boots diskless from a DIM server.
DIM master node The single node, one per distribution, used as the master.
DIM master directory For example, /var/dim/SLES9/master on every DIM server

DIM server setup

The DIM Server is configured with DHCP and NFS. It should have enough disk space to contain the master distribution image, including the software loaded on the image. In addition, it will contain the mount points of all the boot images for each blade connected to the DIM server. Here is the additional information you will need to install and configure the DIM server to manage the blades.

Load the DIM server with the standard SLES 9 distribution; refer to “Appendix A. Software and test tools installed” on page 11. Perform the following tasks.

- Increase the number of NFS kernel threads:

```
/etc/sysconfig/nfs
USE_KERNEL_NFSD_NUMBER=“32”
```

Next, be sure that the interfaces in the DIM Server can provide DHCP capabilities. Your environment may use different adapters.

1. Define DHCP interfaces:

```
/etc/sysconfig/dhcpd
```

```
DHCPD_INTERFACE=“eth0 eth1 myri0”
```

2. Install and start the TFTP server.

   This is included in the SLES distribution, but may not be installed as part of your default install.

3. Increase number of loopback devices:

   ```
   /etc/yaboot.conf
   append max_loop=256
   reboot
   ```

4. Install the additional DIM utilities. We installed DIM version 0.9.6.1:

   ```
   tar -xzf tools_0.9.6-1.tar.gz
   cd tools_0.9.6-1
   make install
   make clean
   ```

5. Install the DIM rpm package:

   ```
   rpm -ivh dim-0.9.6-1.noarch.rpm
   ```

6. To log DIM error messages in syslog:

   ```
   /etc/syslog.conf
   local6.*          /var/log/dim.log
   ```

7. Adjust the shell environment (optional):

   ```
   for bash:
   export DIM_HOME=/opt/dim
   export PATH=$PATH:$DIM_HOME/bin
   export MANPATH=`manpath`:$DIM_HOME/man
   ```

Provide disk space for distributions

Our DIM Server happened to come with eight disk drives. We used the first drive for the operating system, and kept the second as an additional drive for the OS. We mirrored the next two disks and mounted as /var/dim/SLES9 (this is the directory that dim uses to hold the distribution.) We did the same mirror for the next two disks and mounted as /var/dim/RHAT. We reserved the last two disks in case we needed to add to one of the distributions.

BladeCenter management module setup

There are a few settings you must make on the management module interface in order to allow the DIM Server to communicate with the blade chassis, and vice versa.

We configured each management module to allow communication to the respective DIM Server. Because we had three DIM Servers in our environment, there were differences for some chassis. This is done on each chassis.

Enable SNMP by using the telnet command line interface; see Example 1.

Example 1   Enabling SNMP using telnet

```
telnet mm
snmp -T system:mm[1] -a -on
snmp -T system:mm[1] -c1 public
snmp -T system:mm[1] -ca1 set
snmp -T system:mm[1] -c1i1 10.3.0.127
reset -T system:mm[1]
```

Alternatively, you can use the Web front-end for this task; see Example 2.

Example 2 Enabling SNMP using the Web front-end

```
MM-Control
 Æ Network Protocols
SNMPv1 agent enabled
Community Name       Access Type      Host Name or IP Address
-------------------------------------------------------------
                Public               T              129.40.15.201
```

If you test the SNMP connection now to ensure it works properly, it will help with diagnosis later on. It is a simple test that provides immediate feedback:

```
   dim_bcadmin -H mm1 setLocationLed on
   dim_bcadmin -H mm1 showLeds
   dim_bctool  -H mm1 1 info
```

DIM master node setup

The master node is simply one of the blades in your environment. You build the master node manually with the operating system and all necessary settings. If there are known tuning tasks that you normally set, this is the node to set them on. At this time you should also install any default software that you want installed on every node in the complex.

By maintaining a master node you install code, updates, and so on to only one place, and then use the dim commands to push them out to all the nodes in the complex.

1. Change the default runlevel to 3.
2. Stop services that you will not use; see Example 3.

Example 3 Stopping unused services

```
insserv -r alsasound
insserv -r cron
insserv -r cups
insserv -r isdn
insserv -r postfix
insserv -r smbfs
insserv -r splash
insserv -r splash_early
insserv -r splash_late
insserv -r xdm
insserv -r fbset
insserv -r slpd
insserv -r iprdump
insserv -r iprinit
insserv -r iprupdate
insserv -r nfsboot
insserv -r nfs
```

In our environment, we used a /bench1 filesystem as the tmp area for file space. This area is automatically cleaned between user usage. Use /bench1 for file space.

We prepared this space by adding the code shown in Example 4 on page 7 to our /etc/init.d/boot.local.

Example 4 Preparing file space

```
if [ ! -d /bench1 ]; then mkdir /bench1; fi
modprobe ipr
mount -t ext2 /dev/sda2 /bench1
chmod 1777 /bench1
```

This line keeps “df” “mount” happy:

```
echo “ln -fns /proc/mounts /etc/mtab” >> /etc/init.d/boot.local
```

We also made other changes to the boot.local file on our master node, as described here:

- We set the default gateway to ensure communication between nodes.
- We set the IP address for the Myrinet interface using a script we call from the boot.local file. (If we had an adapter that we could put in the DIM Server, we would be able to assign it during the dhcp boot, but we did not have an adapter available.)

  The entry in boot.local is:

  ```
  /usr/sbin/set_myri0
  ```

  See “Appendix B. Listing sample of boot.local” on page 12 for the listing of the script.
- We installed the ssh DIM Server public key (~root/.ssh/authorized_keys).

DIM setup

The following commands are DIM-specific. These commands are part of the DIM Server distribution. They are installed in the /opt/dim/bin directory; we appended this to our path.

Next, set up the DIM master directory. Assuming you have set up your extra disk and mount points as described earlier, this will all take place on the mirrored disk.

1. Set up the DIM master directory (for example, /var/dim/SLES9/master).

   ```
   dim_sync_master -d SLES9 -H suse_master_node
   ```

   The previous command copies the master image over to the DIM Server.
2. Build the DIM images.

   Check the required image size (adjust /opt/dim/config/SLES9/dist.cfg):

   ```
   dim_image -d SLES9 -n
   ```
3. Create images for 14 blades at once.

   In our case, we had the korn shell installed on our DIM Server, so we did this in a “for” loop.

   ```
   for I in 1 tween 14
   do
   dim_image -d SLES9 readonly dim-server[1]-center[1]-blade[$i]
     done
   ```

   This creates the directory structure for your boot mount points on the DIM Server. You should see the var/dim/SLES9/mnt/.

   Make sure there is a directory /tftpboot on your DIM server. If not, issue **mkdir /tftpboot**.
4. Build the DIM network boot image (→ /tftpboot/zImage.SLES9).

   ```
   dim_zimage -d SLES9
   ```

5. Configure the NFS server.

```
dim_nfs add -d SLES9 all
dim_nfs status
dim_nfs start
dim_nfs status
```

6. DHCP server configuration.

   Here is the critical part to a successful install. If you do not get the dim_ip.xml correct, it will not work when using the following commands to build your dhcp file.

   Note: This file must be correct to ensure the proper functioning of the blades on the network.

   a. Adjust the DIM IP configuration (/opt/dim/config/dim_ip.xml)

   b. Add global DHCP options.

   ```
   dim_dhcp add option -O UseHostDeclNames=on:DdnsUpdateStyle=none
   dim_dhcp add option -O DomainName=dim
   dim_dhcp add option -O DomainNameServers=10.2.0.127,10.2.128.127
   ```

   c. Add subnets.

   ```
   dim_dhcp add subnet dim-server[1]-gigabit1
   dim_dhcp add subnet dim-server[1]-service1
   dim_dhcp add subnet dim-server[1]-myrinet1
   ```

   d. Add blades (check option -i).

Next, assign names and addresses for your blades; see Example 5. It was extremely helpful to have the hostnames in a sequential order (that is, blade1, blade2, blade3 … blade n), because the “for” loop assigns the hostnames.

Example 5 Assigning names and addresses to blades

```
dim_dhcp add node -H s01c1b01 -d SLES9 dim-server[1]-center[1]-blade[1]
or 14 blade at once (korn style)
for i in {1..14}; do dim_dhcp add node -d SLES9 -H `printf “b03n$i” 1 1 $i`
dim-server[1]-center[1]-blade[$i]; done
dim_dhcp list host
dim_dhcp restart dhcpd
```

Boot the DIM compute nodes

1. The blades need to be set to boot across the network.

- This can be done from the DIM Server.

  ```
  dim_bbs -H mm1 all network
  ```

- Alternatively, it can be done from the Web interface.

  Blade Tasks → **Configuration**

2. Boot Sequence.
3. Click the blade and select **Network BOOTP** as the first device.
4. Power on the nodes.

   ```
   dim_bctool -H mm1 all on
   dim_bctool -H mm1 all info
   ```

After power on, you can watch the boot sequence from the console. After the LED counting is finished, the blade should find the DIM Server and start loading the zImage.

DIM Server commands for image maintenance

- `dim_sync_master`

  Used to synchronize the master directory /var/dim/<dist>/master
- `dim_sync_image`

  Used to synchronize the DIM images
- `dim_sync_server`

  Used to synchronize multiple DIM Servers

It it possible to synchronize a single file only by using the **-f** option.

Summary

In this paper we described the steps needed to install a small environment of 128 blades. Such an environment could contain many thousands of blades and DIM Servers, requiring many network addresses. In an environment like this, it pays to work through the steps of building the boot servers and defining the network addressing that will be used to run the environment.

The diskless boot environment that we built to support our customers has been very well received in our center. After the DIM Servers were established and the infrastructure was in place, the additional work effort to install the rest of the blades was minimal. This flexible environment provided for updates, rebuilding blades, and adding new blades to the infrastructure, with a bonus in terms of time saved. In addition, we gained the ability to update a few blades, then roll out the updates to the rest of the blades in the infrastructure with a simple command.

Reference material

- *IBM BladeCenter JS21: The POWER of Blade Innovation,* SG24-7273

  http://www.redbooks.ibm.com/abstracts/sg247273.html
- *IBM BladeCenter JS21 Technical Overview and Introduction,* REDP-4130

  http://www.redbooks.ibm.com/abstracts/redp4130.html
- *Linux HPC Cluster Installation*, SG24-6041

  http://www.redbooks.ibm.com/abstracts/sg246041.html
- Linux at IBM Web site

  http://www.ibm.com/linux/
- Linux Documentation Project Web site

  http://www.linuxdoc.org

The author of this Redpaper

Bob Maher is a software engineer in the IBM Industry Solutions and Proof of Concept Center in Poughkeepsie, NY, working with BICoC and pSeries® Commercial Benchmarks. He has more than 18 years of IT experience working at IBM. His areas of expertise include Operating System Performance, WebSphere®, Portal Server, and Lotus® Notes® Performance

analysis and design and implementation on the System p™ platform. Bob has led multiple performance benchmarking and capacity planning projects, including the JS21 Diskless boot HPC Cluster implementation in the System P benchmark center.

Acknowledgements

Thanks to the following people for their help in producing this Redpaper:

Lawrence P. Diehl Jr., IBM STG Operations Lab Technician

Mike Ebbers, IBM International Technical Support Organization, Poughkeepsie, NY

Tom Henebery, IBM Industry Solutions and Proof of Concept Center

Peter Morjan, IBM DIM Server Developer

Michael Norton, IBM Benchmark Center Hardware and Network Support

Julie Peet, IBM Team Lead

Greg Rodgers, Program Director NextGen Blade Clusters

Giuseppe Sarrica, IBM STG Industry Solutions and High Performance Computing Project Manager

Barry Speilberg, IBM Industry Solutions and Proof of Concept Center

Nick Venidis, IBM Benchmark and Infrastructure Support

Appendix A. Software and test tools installed

The following is a list of the software and test tools we installed on our environment.

- SUSE SLES9 OS - update 3
- IBM Java™ - IBMJava2-JRE-1.4.2-0.55
- Virtual age C compiler V8.0.0
- XL Fortran V 10.1.0.0
- Lapi - lapi_ppc_64bit_IP_sles900-2.4.0.0-0614a
- ESSL & PESSL
  - essl.rte-4.2.2-3
  - pessl.common-3.2.0-0
- Bison - bison-1.875-51.4
- Flex - flex-64bit-9-200407011606
- MPI Wrappers - mpi-wrappers-1.0.0-7
- MPICH
  - mpich-gm-1.2.6..14b-2
  - mpich-mx-1.2.7..1-2
- GPFS - gpfs.docs-2.3.0-11
- Load Leveler - LoadL-full-SLES9-PPC64-3.3.2.0-0
- IBM Parallel environment
  - IBM_pe_license-4.2.0.0-0614a
  - ppe_ppc_base_32bit_sles900-4.2.0.0-0614a
  - ppe_ppc_64bit_sles900-4.2.0.0-0614a

Appendix B. Listing sample of boot.local

Example 6   boot.local sample

```
#! /bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany.  All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1996
#         Burchard Steinbild, 1996
#! /bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany.  All rights reserved.
#
# Author: Werner Fink <werner@suse.de>, 1996
#         Burchard Steinbild, 1996
#
# /etc/init.d/boot.local
#
# script with local commands to be executed from init on system startup
#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
#

route add default gw 129.40.116.254

rm -f /etc/mtab; ln -s /proc/mounts /etc/mtab

if [ ! -d /bench1 ]; then mkdir /bench1; fi
modprobe ipr
if [ `hostname -s` != b03g01 ] | [ `hostname -s` != b03n01 ]; then mount -t ext2
/dev/sda2 /bench1; fi

chmod 1777 /bench1

/usr/sbin/set_myri0

/usr/lpp/mmfs/bin/mmstartup
```

Appendix C. Listing sample of dhcpd.conf

Example 7 dhcpd.conf sample

```
## DIM START
## This section is managed by DIM (dim_dhcp $Rev: 600 $).
## Don't touch.
##
## Last update : 2006-04-25 13:26:38
##

ddns-update-style none;
option domain-name “pbm.ihost.com”;
option domain-name-servers 129.40.106.1;
use-host-decl-names on;

subnet 129.40.2.0 netmask 255.255.255.0 {
    # dim-server[3]-gigabit1
}
subnet 129.40.116.0 netmask 255.255.255.0 {
    # dim-server[3]-gigabit2
}
subnet 129.40.102.0 netmask 255.255.255.0 {
    # dim-server[3]-myrinet1
}
subnet 129.40.15.0 netmask 255.255.255.0 {
    # dim-server[3]-service1
}
host dim-129-40-2-99 {
    # dim-server[3]-center[1]-blade[1]-gigabit1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.2.99;
    option host-name “b03g99”;
    hardware ethernet 00:11:25:c9:19:c0;
    next-server 129.40.2.203;
    option root-path
“129.40.2.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[1]”;
}
host dim-129-40-116-99 {
    # dim-server[3]-center[1]-blade[1]-gigabit2
    filename “zImage.SLES9bm”;
    fixed-address 129.40.116.99;
    option host-name “b03n99”;
    hardware ethernet 00:11:25:c9:19:c1;
    next-server 129.40.116.203;
    option root-path
“129.40.116.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[1]”;
}
host dim-129-40-102-99 {
    # dim-server[3]-center[1]-blade[1]-myrinet1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.102.99;
    option host-name “b03m99”;
    hardware ethernet 00:60:dd:47:f5:9e;
    next-server 129.40.102.203;
    option root-path
“129.40.102.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[1]”;
}
host dim-129-40-2-100 {
    # dim-server[3]-center[1]-blade[2]-gigabit1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.2.100;
    option host-name “b03g100”;
    hardware ethernet 00:11:25:c9:1a:1c;
    next-server 129.40.2.203;
    option root-path
“129.40.2.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[2]”;
}
host dim-129-40-116-100 {
    # dim-server[3]-center[1]-blade[2]-gigabit2
    filename “zImage.SLES9bm”;
    fixed-address 129.40.116.100;
    option host-name “b03n100”;
    hardware ethernet 00:11:25:c9:1a:1d;
    next-server 129.40.116.203;
    option root-path
“129.40.116.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[2]”;
}
host dim-129-40-102-100 {
    # dim-server[3]-center[1]-blade[2]-myrinet1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.102.100;
    option host-name “b03m100”;
    hardware ethernet 00:60:dd:47:f5:5b;
    next-server 129.40.102.203;
    option root-path
“129.40.102.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[2]”;
}
host dim-129-40-2-101 {
    # dim-server[3]-center[1]-blade[3]-gigabit1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.2.101;
    option host-name “b03g101”;
    hardware ethernet 00:11:25:c9:19:06;
    next-server 129.40.2.203;
    option root-path
“129.40.2.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[3]”;
}
host dim-129-40-116-101 {
    # dim-server[3]-center[1]-blade[3]-gigabit2
    filename “zImage.SLES9bm”;
    fixed-address 129.40.116.101;
    option host-name “b03n101”;
    hardware ethernet 00:11:25:c9:19:07;
    next-server 129.40.116.203;
    option root-path
“129.40.116.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[3]”;
}

host dim-129-40-102-101 {
    # dim-server[3]-center[1]-blade[3]-myrinet1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.102.101;
    option host-name “b03m101”;
    hardware ethernet 00:60:dd:47:f5:33;
    next-server 129.40.102.203;
    option root-path
“129.40.102.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[3]”;
}
host dim-129-40-2-102 {
    # dim-server[3]-center[1]-blade[4]-gigabit1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.2.102;
    option host-name “b03g102”;
    hardware ethernet 00:11:25:c9:19:46;
    next-server 129.40.2.203;
    option root-path
“129.40.2.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[4]”;
}
host dim-129-40-116-102 {
    # dim-server[3]-center[1]-blade[4]-gigabit2
    filename “zImage.SLES9bm”;
    fixed-address 129.40.116.102;
    option host-name “b03n102”;
    hardware ethernet 00:11:25:c9:19:47;
    next-server 129.40.116.203;
    option root-path
“129.40.116.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[4]”;
}
host dim-129-40-102-102 {
    # dim-server[3]-center[1]-blade[4]-myrinet1
    filename “zImage.SLES9bm”;
    fixed-address 129.40.102.102;
    option host-name “b03m102”;
    hardware ethernet 00:60:dd:47:f5:9f;
    next-server 129.40.102.203;
    option root-path
“129.40.102.203:/var/dim/SLES9bm/mnt/dim-server[3]-center[1]-blade[4]”;
}
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Redpaper

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

BladeCenter®
IBM®
Lotus®
Lotus Notes®
Notes®
POWER™
pSeries®
Redbooks™
Redbooks (logo) ™
System p™
WebSphere®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.