Redbooks Paper

Robert Brenneman

Cloning FCP-attached SCSI SLES9 Linux for zSeries Systems

Introduction

This Redpaper describes a procedure to clone a root file system residing on a Fibre Channel Protocol (FCP) attached to a small computer system interface (SCSI) disk. Cloning SCSI disks that act as a Linux® root file system requires special processing (compared to cloning ECKD™ DASD):

- An initial ramdisk, containing the zfcp device driver, is required to initial program load (IPL) from SCSI. The ramdisk also contains the FCP device mapping needed to access the root file system on the SCSI disk.
- When the file system is copied, a new initial ramdisk must be created on the clone disk. This ramdisk must map the clone SCSI disk as the root file system (and not the file system on the original master SCSI disk).

We illustrate the steps using a SUSE SLES9 Linux distribution running as a guest under z/VM®. The procedure also applies to Linux running in an LPAR on zSeries®. Two FCP devices are required:

- One device accesses the master Logical Unit Number (LUN).

  This contains the root file system for an existing SLES9 installation. In the examples that follow, the master LUN is assigned to device address A000.
- One device accesses the clone LUN.

  The master LUN is copied to the clone LUN. This forms the root file system for another SLES9 installation. In the examples that follow, the clone LUN is assigned to device address A001.

These devices can be assigned to:

- Separate FCP adapters
- Separate CHPIDs on the same adapter
- Two devices on the same CHPID (if using z/VM).

Two devices are needed because the ramdisk created on the clone LUN maps both the master and clone LUNs. In order to IPL from the clone LUN, the master LUN must be made unavailable to the Linux system (using the **VARY OFFLINE** or **DETACH** commands). After the IPL process, the ramdisk on the clone LUN is rebuilt to remove references to the master LUN.

Note: If the clone Linux system is to run in an LPAR, the two FCP devices *must* be assigned to separate CHPIDs. This allows the device used to access the master LUN that you bring offline when you IPL from the clone LUN.

Add the clone LUN to the master Linux system

1. Use the following command to ensure that the device used to access the clone LUN is online.

   ```
   # echo 1 > /sys/bus/ccw/drivers/zfcp/0.0.a001/online
   ```

   If the device is not already online, the following messages appear on the console:

   ```
   scsi1 : zfcp
   Nov  2 18:30:39 ltic0018 kernel: scsi1 : zfcp
   ```

2. Add the Worldwide Port Name (WWPN) used to access the new LUN to the device.
3. Add the new LUN behind the WWPN as shown in Figure 1.

```
# cd /sys/bus/ccw/drivers/zfcp/0.0.a001
# echo 0x5005076300cdafc4 > port_add
# cd 0x5005076300cdafc4
# ls
.   d_id          failed      scsi_id  unit_add     wwnn
..  detach_state  in_recovery  status   unit_remove
# echo 0x5401000000000000 > unit_add
```

Figure 1 Add the new WWPN and LUN

Note: In this example, LUN 0x5401000000000000 is accessed using WWPN 0x5005076300cdafc4. First add the WWPN using the port_add interface of the A001 device. Next, add the LUN using the unit_add interface of the 0x5005076300cdafc4 WWPN.

After the LUN is added, the messages in Figure 2 on page 3 appear on the console.

```
 Vendor: IBM       Model: 2105800           Rev: .115
  Type:   Direct-Access                     ANSI SCSI revision: 03
Oct 29 19:44:56 ltic0018 kernel:   Vendor: IBM       Model: 2105800           Re
v: .115
SCSI device sdb: 19531264 512-byte hdwr sectors (10000 MB)
SCSI device sdb: drive cache: write back
 sdb:Oct 29 19:44:56 ltic0018 kernel:   Type:   Direct-Access
   ANSI SCSI revision: 03
Oct 29 19:44:56 ltic0018 kernel: SCSI device sdb: 19531264 512-byte hdwr sectors
 (10000 MB)
Oct 29 19:44:56 ltic0018 kernel: SCSI device sdb: drive cache: write back
 unknown partition table
Attached scsi disk sdb at scsi1, channel 0, id 1, lun 0
Attached scsi generic sg1 at scsi1, channel 0, id 1, lun 0,  type 0
Oct 29 19:44:56 ltic0018 kernel:  sdb: unknown partition table
Oct 29 19:44:56 ltic0018 kernel: Attached scsi disk sdb at scsi1, channel 0, id
2, lun 0
Oct 29 19:44:56 ltic0018 kernel: Attached scsi generic sg1 at scsi1, channel 0,
id 2, lun 0,  type 0
```

Figure 2 Console messages when adding the new LUN

Copy the master root file system to the clone LUN

1. Use the **dd** command to copy the root file system of the master Linux system to the new SCSI disk:

   ```
   # dd if=/dev/sda of=/dev/sdb bs=512
   19531264+0 records in
   19531264+0 records out
   ```

   This alters the partition table of /dev/sdb. In order to see the partitions on the /dev/sdb device, the system must refresh its view of the device partition table.

2. Force Linux to re-read the /dev/sdb partition table using the **fdisk** command, as shown in Figure 3 on page 4.

```
# fdisk /dev/sdb

The number of cylinders for this disk is set to 9536.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdb: 10.0 GB, 10000007168 bytes
64 heads, 32 sectors/track, 9536 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        9217     9438192   83  Linux
/dev/sdb2            9218        9536      326656   82  Linux swap

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Figure 3 Refresh the /dev/sdb partition table using the **fdisk** *command*

In this example, the partition table is not changed. Simply writing no changes to the disk (using the **w** command) causes Linux to refresh its view of the partition table.

Note: You can copy the master file system to the clone LUN outside of Linux (for example, using IBM® Enterprise Storage System Copy Services). In this case, when you IPL Linux and the clone LUN attached to the guest, Linux immediately sees the partitions on the clone disk.

Configure the clone root file system

1. Mount the clone disk on the /mnt directory of the Linux master.
2. Mount /mnt/proc and /mnt/sys to enable the **mkinitrd** command to work in a chroot environment, as shown in Figure 4.

```
# mount /dev/sdb1 /mnt
# mount -t proc proc /mnt/proc
# mount -t sysfs sysfs /mnt/sys
# mount
/dev/sda1 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
tmpfs on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sdb1 on /mnt type reiserfs (rw)
proc on /mnt/proc type proc (rw)
sysfs on /mnt/sys type sysfs (rw)
```

Figure 4 Mount the clone SCSI disk

3. Execute the **chroot** command to the /mnt directory, and make the required changes to the clone LUN, as shown in Figure 5.

```
# cd /
# chroot /mnt /bin/bash
# touch THIS_IS_THE_CLONE
# ls
.   THIS_IS_THE_CLONE  boot  etc   lib  opt   root  srv  tmp  var
..  bin                dev   home  mnt  proc  sbin  sys  usr
# cd /etc
# touch THIS_IS_THE_CLONE
# cd sysconfig/hardware/
# ls
.   config                        hwcfg-zfcp-bus-ccw-0.0.a000  skel
..  hwcfg-qeth-bus-ccw-0.0.0600   scripts
```

Figure 5 Execute the chroot to the /mnt directory

Note: The hwcfg-zfcp files in the /etc/sysconfig/hardware directory are used to configure FCP devices. The full file name indicates the FCP device to configure. In this example, the hwcfg-zfcp-bus-ccw-0.0.a000 file contains configuration parameters for SCSI disks accessed through the FCP device at virtual address A000.

The contents of the hwcfg-zfcp-bus-ccw-0.0.a000 file are shown in Figure 6.

```
#!/bin/sh
#
# hwcfg-zfcp-bus-ccw-0.0.a000
#
# Configuration for the zfcp adapter at CCW ID 0.0.a000
#

STARTMODE="auto"
MODULE="zfcp"
MODULE_OPTIONS=""
MODULE_UNLOAD="yes"

# Scripts to be called for the various events.
# If called manually the event is set to 'up'.
SCRIPTUP="hwup-ccw"
SCRIPTUP_ccw="hwup-ccw"
SCRIPTUP_scsi_host="hwup-zfcp"
SCRIPTDOWN="hwdown-scsi"
SCRIPTDOWN_scsi="hwdown-zfcp"

# Configured zfcp disks
ZFCP_LUNS="0x5005076300ceafc4:0x5400000000000000"
```

Figure 6 Contents of the /etc/sysconfig/hardware/hwcfg-zfcp-bus-ccw-0.0.a000 file

4. Because the /mnt file system is a clone of a SCSI root file system, the existing hwcfg-zfcp-bus-ccw-0.0.a000 file configures the master LUN. Rename hwcfg-zfcp-bus-ccw-0.0.a000 to hwcfg-zfcp-bus-ccw-0.0.a001 (configuring the FCP device at virtual address A001).

5. Modify the contents of hwcfg-zfcp-bus-ccw-0.0.a001 to point to the WWPN and LUN of the clone disk. The new file contents are shown in Figure 7 with the modifications highlighted.

```
#!/bin/sh
#
# hwcfg-zfcp-bus-ccw-0.0.a001
#
# Configuration for the zfcp adapter at CCW ID 0.0.a001
#

STARTMODE="auto"
MODULE="zfcp"
MODULE_OPTIONS=""
MODULE_UNLOAD="yes"

# Scripts to be called for the various events.
# If called manually the event is set to 'up'.
SCRIPTUP="hwup-ccw"
SCRIPTUP_ccw="hwup-ccw"
SCRIPTUP_scsi_host="hwup-zfcp"
SCRIPTDOWN="hwdown-scsi"
SCRIPTDOWN_scsi="hwdown-zfcp"

# Configured zfcp disks
ZFCP_LUNS="0x5005076300cdafc4:0x5401000000000000"
```

Figure 7 Contents of the /etc/sysconfig/hardware/hwcfg-zfcp-bus-ccw-0.0.a001 file

6. Create a new initial ramdisk on the clone disk using the **mkinitrd** command, as shown in Figure 8. This ramdisk uses the new hardware configuration.

```
# mkinitrd
Root device:    /dev/sda1 (mounted on / as reiserfs)
Module list:    reiserfs sd_mod zfcp

Kernel image:   /boot/image-2.6.5-7.97-s390
Initrd image:   /boot/initrd-2.6.5-7.97-s390
Shared libs:    lib/ld-2.3.3.so lib/libc.so.6 lib/libselinux.so.1
Modules:        kernel/fs/reiserfs/reiserfs.ko kernel/drivers/scsi/scsi_mod.ko
kernel/drivers/scsi/sd_mod.ko kernel/drivers/s390/cio/qdio.ko
kernel/drivers/s390/scsi/zfcp.ko
zfcp HBAs:      0.0.a000 0.0.a001
zfcp disks:
                0.0.a000:0x5005076300ceafc4:0x5400000000000000
                0.0.a001:0x5005076300cdafc4:0x5401000000000000

initrd updated, zipl needs to update the IPL record before IPL!
```

Figure 8 Update the initial ramdisk on the clone LUN

Note: The **mkinitrd** command created a ramdisk using both the A000 (master root) and A001 (clone root) FCP devices. It is not possible to specify only the A001 device (**mkinitrd** adds all active FCP devices). You will remove the A000 device reference later when you IPL the clone system.

7. Execute the **zipl** command to pick up the new initial ramdisk on the clone disk. See Figure 9.

```
# cd /boot
# zipl -V
Using config file '/etc/zipl.conf'
Target device information
  Device......................: 08:10
  Partition...................: 08:11
  Device name.................: sdb
  Type........................: disk partition
  Disk layout.................: SCSI
  Geometry - heads............: 64
  Geometry - sectors..........: 32
  Geometry - cylinders........: 9536
  Geometry - start............: 32
  File system block size......: 4096
  Physical block size.........: 512
  Device size in physical blocks..: 18876384
Building bootmap '/boot/zipl/bootmap'
Adding IPL section 'ipl' (default)
  kernel image......: /boot/image at 0x10000
  kernel parmline...: 'root=/dev/sda1 selinux=0 TERM=dumb elevator=cfq' at 0x1000
  initial ramdisk...: /boot/initrd at 0x800000
Preparing boot device: sdb.
Detected SCSI PCBIOS disk layout.
Writing SCSI master boot record.
Syncing disks...
Done.
```

*Figure 9 Run **zipl** on the clone LUN*

8. IPL the clone disk.
9. Make the necessary customizations to the clone disk. For instances, you can change the host name and IP address.
10. After you make all of the changes, exit the chroot environment, and unmount the clone disk as shown in Figure 10.

```
# exit
exit
ltic0018:~ # ls /
.   bin   dev  home  mnt  proc  sbin  sys  usr
..  boot  etc  lib   opt  root  srv   tmp  var
# umount /mnt/proc
# umount /mnt/sys
# umount /mnt
# mount
/dev/sda1 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
tmpfs on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
```

Figure 10 Exit the chroot environment and unmount the clone disk

IPL from the clone disk

Shutdown the system and IPL from the clone disk, which is shown in Figure 11.

- Use the SET LOADDEV command to point to the WWPN and LUN of the clone disk.
- Use the values specified when creating the initial ramdisk on the clone disk.
- Ensure that the master LUN device is offline or detached from the VM guest.

```
SET LOADDEV PORTNAME 50050763 00CDAFC4 LUN 54010000 00000000
DET A000
FCP A000 DETACHED
I A001
HCPLDI2816I Acquiring the machine loader from the processor controller.
HCPLDI2817I Load completed from the processor controller.
HCPLDI2817I Now starting machine loader version 0001.
MLOEVL012I: Machine loader up and running (version 0.13).
MLOPDM003I: Machine loader finished, moving data to final storage location.
Linux version 2.6.5-7.97-s390 (geeko@buildhost) (gcc version 3.3.3 (SuSE Linux))
 #1 SMP Fri Jul 2 14:21:59 UTC 2004
We are running under VM (31 bit mode)
This machine has an IEEE fpu
On node 0 totalpages: 131072
  DMA zone: 131072 pages, LIFO batch:16
  Normal zone: 0 pages, LIFO batch:1
  HighMem zone: 0 pages, LIFO batch:1
Built 1 zonelists
Kernel command line: root=/dev/sda1 selinux=0 TERM=dumb elevator=cfq
PID hash table entries: 4096 (order 12: 32768 bytes)
CKRM Initialization
...... Initializing ClassType<taskclass> ........
...... Initializing ClassType<socket_class> ........
CKRM Initialization done
Memory: 511616k/524288k available (3052k kernel code, 0k reserved, 1299k data, 9
2k init)
Calibrating delay loop... 1992.29 BogoMIPS
Security Scaffold v1.0.0 initialized
SELinux:  Disabled at boot.
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Detected 1 CPU's
Boot cpu address  0
cpu 0 phys_idx=0 vers=FF ident=01AC7A machine=2086 unused=8000
Brought up 1 CPUs
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 1317k freed
debug: Initialization complete
NET: Registered protocol family 16
VFS: Disk quotas dquot_6.5.1
```

Figure 11 IPL the clone disk

You may see messages indicating device A000 is not being used or is inoperable. This is normal and acceptable (the device was detached from the guest). In Figure 12 on page 9, we execute the **mkinitrd** and **zipl** commands to remove references to the A000 device from the initial startup scripts.

```
# mkinitrd
Root device:    /dev/sda1 (mounted on / as reiserfs)
Module list:    reiserfs sd_mod zfcp

Kernel image:   /boot/image-2.6.5-7.97-s390
Initrd image:   /boot/initrd-2.6.5-7.97-s390
Shared libs:    lib/ld-2.3.3.so lib/libc.so.6 lib/libselinux.so.1
Modules:        kernel/fs/reiserfs/reiserfs.ko kernel/drivers/scsi/scsi_mod.ko
kernel/drivers/scsi/sd_mod.ko kernel/drivers/s390/cio/qdio.ko
kernel/drivers/s390/scsi/zfcp.ko
zfcp HBAs:      0.0.a001
zfcp disks:
                0.0.a001:0x5005076300cdafc4:0x5401000000000000

initrd updated, zipl needs to update the IPL record before IPL!
# zipl -V
Using config file '/etc/zipl.conf'
Target device information
  Device......................: 08:00
  Partition...................: 08:01
  Device name.................: sda
  Type........................: disk partition
  Disk layout.................: SCSI
  Geometry - heads............: 64
  Geometry - sectors..........: 32
  Geometry - cylinders........: 9536
  Geometry - start............: 32
  File system block size......: 4096
  Physical block size.........: 512
  Device size in physical blocks..: 18876384
Building bootmap '/boot/zipl/bootmap'
Adding IPL section 'ipl' (default)
  kernel image......: /boot/image at 0x10000
  kernel parmline...: 'root=/dev/sda1 selinux=0 TERM=dumb elevator=cfq' at 0x1000
  initial ramdisk...: /boot/initrd at 0x800000
Preparing boot device: sda.
Detected SCSI PCBIOS disk layout.
Writing SCSI master boot record.
Syncing disks...
Done.
```

Figure 12 Create an initial ramdisk without the master disk

11. IPL again to ensure that everything works correctly.

The team that wrote this Redpaper

This Redpaper was produced by a specialist working at the International Technical Support Organization, Poughkeepsie Center.

Robert Brenneman Robert Brenneman is a Software Engineer at the Linux Test and Integration Center in Poughkeepsie, NY.

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 illustrates 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. 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 IBM's application programming interfaces.

This document created or updated on December 7, 2004.

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. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

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

ECKD™	IBM®	z/VM®
@server®	OS/2®	zSeries®
ibm.com®	Redbooks (logo) ™	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Micorosoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered 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, and service names may be trademarks or service marks of others.