Linux®

Snapshot™ Records and LUN Resizing in a SAN Environment

Network Appliance | 1/27/2003

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Abstract

This technical report contains steps required to perform common operations on a Linux host and a Network Appliance™ filer utilizing either the iSCSI protocol or FCP. Specifically we cover the creation of Snapshot records on the filer that are consistent and reliable from the perspective of the host. Further, we cover the use of the snaprestore command to restore a LUN to a previous state using Snapshot records. We also cover the process of increasing the size of a NetApp filer LUN so that the space is available on the Linux host. All of these operations will eventually be provided by the SnapDrive™ product when it ships on the Linux platform. This technical report is an aid to customers who need the functionality in before the full-fledged product is available.

Network Appliance Inc.

Table of Contents

1. Purpose

This technical report contains steps required to perform common operations on a Linux host and a Network Appliance filer utilizing either the iSCSI protocol or FCP. Specifically, we cover the creation of Snapshot records on the filer that are consistent and reliable from the perspective of the Linux host. Further, we cover the use of the snaprestore command to restore a LUN to a previous state using Snapshot records. We also cover the process of increasing the size of a NetApp filer LUN so that the space is available on the Linux host. All of these operations will eventually be provided by the SnapDrive product when it ships on the Linux platform. However, this technical report assumes that SnapDrive is not used and is thus suitable for customers who cannot or choose not to use SnapDrive for any reason, or for customers who wish to implement these features on Linux prior to the shipment of SnapDrive for Linux.

2. Overview

Network Appliance, Inc. provides the iSCSI protocol and Fibre Channel protocol (FCP) on its current version (6.4 and forward) of the Data ONTAP™ operating system. Network Appliance also provides the following in terms of SAN support on Linux:

- A Host Attach Kit for FCP
- Support for an iSCSI software initiator, as well as certain iSCSI hardware initiators under the open iSCSI program

More information on the FCP Host Attach Kit for the Linux platform can be found here. Further, information on iSCSI initiator support for Linux can be found here.

To facilitate the adoption of these protocols and to expose key functionality provided by NetApp filers in the area of Snapshot records, NetApp intends to develop a SnapDrive client product for Linux.

However, presently, the SnapDrive product is only available for Windows®. More information on SnapDrive can be found here. In general, SnapDrive enables the following features:

1. Creating a Snapshot record of a LUN containing a file system
2. Performing a snaprestore of a LUN containing a file system
3. Resizing a partition with a file system that is stored on a NetApp LUN
4. Importing an existing LUN containing a file system onto a Linux host

Pending the shipment of the SnapDrive product for Linux, this technical report fills in the gaps by providing the same (or as close to the same as possible) functionality as SnapDrive, using scripts, the NetApp command line interface, and functionality embedded in the Linux operating system.

While this integration guide contains detailed instructions and procedures for the SAN protocols and its use with Linux and NetApp filers, it is not an exhaustive configuration report for any of these technologies alone. The Network Appliance internal and external Web sites contain numerous reports on these subjects, all of which the reader is encouraged to examine.

3. Assumptions

We ran with Red Hat Advanced Server version 2.1. We used the iSCSI Linux Initiator Support Kit 1.0 (see the Setup Guide for more information) and the FCP Host Attach Kit (see the Installation and Setup Guide for more information). Any version of a NetApp filer and Data ONTAP that supports iSCSI or FCP should work with the techniques presented in this technical report. See the SAN Compatibility Matrix for more information. Other details of our configuration include:

- The filer name was data.
- The name of the Linux host was borg2of6.
- The location of the LUN on the filer was /vol/linuxblocks/linuxblockslun0.
- Under the iSCSI software initiator, this LUN was exposed to the Linux host at /dev/iscsi/bus0/target0/lun0. Under FCP, it was exposed to the Linux host as /dev/sdb.
- This LUN was mounted on the Linux host at /lun0.
- The LUN was formatted as ext3 (ext2 with journaling).
- Both the host and the LUN are mapped to an igroup called linuxigroup.

4. Creating Snapshot Records on a Filer Using iSCSI or FCP Under Linux

Snapshot records can be supported within Linux, provided that the following best practices are carefully followed. The principal requirement is that you must flush the file system buffer cache yourself. This is accomplished on Linux using the sync command. Unlike with other operating systems resembling UNIX®, sync on Linux is synchronous. The standard man page for this command on Linux states:

> According to the standard specification (e.g., SVID), **sync()** schedules the writes, but may return before the actual writing is done. However, since version 1.3.20 Linux does actually wait.

Accordingly, we can rely on the sync command to flush all pending writes to disk prior to creating a Snapshot record. By combining the sync command with the journaling feature of the ext3 file system, we can achieve usable Snapshot support. This technique is used in this technical report. (Note: If writes

occur to the file system after the sync is issued, but before the Snapshot record is created, then an fsck of the file system may be required after snaprestore.)

If this recommendation is followed, creating a Snapshot record of a filer LUN mounted on a Linux host is a secure, reliable operation. The remainder of this section contains detailed procedures for using Snapshot records using this method.

To create a Snapshot record of a LUN stored on a NetApp filer mounted on a Linux host, follow this procedure:

1. Run sync on the Linux host:

```
[root@borg2of6 etc]# sync
[root@borg2of6 etc]#
```

2. Create a Snapshot record on the filer:

```
data> snap create linuxblocks linuxblockssnap
data> snap list linuxblocks
Volume linuxblocks
working...

  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  Dec 02 15:48  linuxblockssnap

data>
```

At this point, you can use SnapMirror®, SnapVault™, tape backup, or other backup devices to back up the LUN from within the Snapshot record. The techniques required to accomplish this are beyond the scope of this integration guide. For more information on this issue, see Data Protection Strategies for Network Appliance Filers by Nicholas Wilhelm-Olsen et al.

The following table summarizes the process of creating a Snapshot record of a filer LUN mounted on a Linux host:

Operation	Purpose
1. Run sync on the Linux host.	Flushes the file system buffer cache.
2. Create a Snapshot record on the filer.	Creates a read-only image of the entire ext3 file system as of the moment in time when the Snapshot record is created.

5. Restoring a LUN to a Prior State Using SnapRestore Under Linux

An ext3 file system that has been backed up using the process shown in the preceding subsection may be restored in a variety of ways:

1. You can create a writable LUN from within the Snapshot record using the “lun create –b” command. Then you can mount the LUN over either iSCSI or FCP and use operating system commands from within Linux to restore the data. (You can also use the “lun clone” command

on a LUN created with “lun create –b” to make the backup LUN independent of the Snapshot record. For more information on this issue, see the NetApp System Administration Guide.) While this works, it is slow. When the LUN created with “lun create –b” is mounted, the data must be copied from this LUN to the original LUN. In the process, the data must be copied from the filer to the Linux host and back to the filer again. Also, there often is no reason to do this, as the entire file system must be replaced anyway. Therefore, this technique is only used when a small fraction of the total data must be restored.

2. You can perform a file-level snaprestore command on the filer to restore the LUN. This technique is very fast and also only reverts the data contained in the LUN. The remainder of the filer volume is unaffected. This is the most common way of reverting a LUN for this reason.
3. You can perform a volume-level snaprestore. This is also a very rapid operation. However, all data in the filer volume is reverted to the Snapshot record. This may be undesirable or even destructive. Extreme caution should be exercised when performing a volume-level snaprestore.

The rest of this subsection assumes that you will restore the database by performing a file-level snaprestore.

In the process of performing a file-level snaprestore, data representing changes to the LUN after creating the Snapshot record will be lost, of course. That is the nature of a snaprestore.

1. Unmount the file system. The method to accomplish this depends on the protocol. Under either FCP or iSCSI using a hardware initiator, a simple “umount” of the file system’s mount point will suffice. Under the iSCSI software initiator, things are a bit more complicated. The easiest way to do this is to comment out the mount point for the LUN in the “/etc/fstab.iscsi” file and then run “service iscsi restart.”
2. Next you should offline the LUN. This will allow you to revert the LUN without affecting the Linux host.

```
data> lun offline /vol/linuxblocks/linuxblockslun0
data> lun show /vol/linuxblocks/linuxblockslun0 –v
        /vol/linuxblocks/linuxblockslun0     10g (10737418240)   (r/w,
offline, mapped)
                Serial#: OdDeR3jXsdEd
                Share: none
                Space Reservation: enabled
                Multiprotocol Type: linux
                Maps: linuxigroup=0
```

3. Scan the storage protocol’s bus to ensure that the LUN has disappeared. For the iSCSI software initiator, this would be:

```
[root@borg2of6 etc]# service iscsi restart
Stopping iSCSI: sync umount sync iscsid iscsi
   <some lines omitted>
[root@borg2of6 etc]# ls -l /dev/iscsi/bus0/target0
total 0
```

Note that there are no LUNs under /dev/iscsi/bus0/target0. This indicates that the LUN has vanished from the Linux host. This is normal.

Under FCP, the commands to rescan the SCSI bus would be:

```
[root@borg2of6 etc]# modprobe –r qla2300
[root@borg2of6 etc]# modprobe qla2300
```

Note: Using “modprobe” in this way will unload and reload the Qlogic HBA driver. This requires that all I/O to any LUN accessed using this driver cease, and the data is briefly offline while this operation runs. A less intrusive approach to rescanning the SCSI bus can be found here. This script is available by default under SuSE Linux. On Red Hat Linux, it must be installed.

Running the “fdisk –l” command will show that /dev/sdb has vanished. This indicates that the LUN is no longer connected to the Linux host:

```
[root@borg2of6 etc]# fdisk –l

Disk /dev/sda: 255 heads, 63 sectors, 2213 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1      2082  16723633+  83  Linux
/dev/sda2          2083      2213   1052257+  82  Linux swap
```

iSCSI hardware initiators may have other methods of scanning their bus. Check the documentation of your iSCSI hardware HBA for details.

4. At this point, the LUN has been completely disconnected from the Linux host, and you have confirmed this fact. You can now snaprestore the LUN:

```
data> snap restore -t file -s linuxblockssnap /vol/linuxblocks/linuxblockslun0

WARNING! This will restore a file from a snapshot into the active
filesystem.  If the file already exists in the active filesystem,
it will be overwritten with the contents from the snapshot.

Are you sure you want to do this? Y

You have selected file /vol/linuxblocks/linuxblockslun0, snapshot
linuxblockssnap

Proceed with restore? Y
Tue Dec  2 16:41:25 EST [vdisk_admin:notice]: [-889139140, 112, 234760497]
snap restore: started
data> Tue Dec  2 16:41:27 EST [vdisk_admin:notice]: [-889139140, 112,
234760497] snap restore: completed
```

5. Now, online the LUN again:

```
data> lun online /vol/linuxblocks/linuxblockslun0
data> lun show -v /vol/linuxblocks/linuxblockslun0
        /vol/linuxblocks/linuxblockslun0     10g (10737418240)   (r/w, online,
mapped)
                Serial#: OdDeR3jXsdEd
                Share: none
                Space Reservation: enabled
                Multiprotocol Type: linux
                Maps: linuxigroup=0
```

6. Rescan the storage bus from within the Linux host. See step 3 in this section for details on how to accomplish this.
7. Remount the file system. Under the iSCSI software initiator, this would require uncommenting the entry in the /etc/fstab.iscsi file and then running “service iscsi restart” again. Note that you may see a warning concerning the journal being recovered:

```
[root@borg2of6 etc]# service iscsi restart
   <some lines omitted>

[root@borg2of6 etc]# /dev/iscsi/bus0/target0/lun0/part1: recovering journal
/dev/iscsi/bus0/target0/lun0/part1: clean, 11/1310720 files, 49345/2621436
blocks
```

   For FCP or an iSCSI hardware HBA, a simple mount command on the mount point will suffice.

Your Linux file system is now available, but data entered after the Snapshot record was created has been lost. If necessary, you will need to reenter this data.

The following table summarizes the process of performing a snaprestore of a filer LUN mounted on a Linux host:

Operation	Purpose
1. Unmount the file system stored on the LUN.	Prevents abrupt disruption of the file system when snaprestore occurs. Without unmounting the file system, this is not a secure or reliable operation.
2. Offline the LUN.	Takes the LUN out of the Linux host’s sphere of control.
3. Scan the SCSI bus.	Ensures that the Linux host sees that the LUN has vanished.
4. Snaprestore the LUN.	Restores the data on the LUN to the state when the Snapshot record was created.
5. Online the LUN.	Allows the Linux host to see the LUN again.
6. Scan the SCSI bus.	Ensures that the Linux host sees the LUN again.
7. Remount the file system.	Allows the file system to be open for reading and writing by users. The data in the file system should now be that which existed when the Snapshot record was created.

6. Resizing a Partition Stored on a NetApp Filer LUN Under Linux

The following procedure will allow you to resize a NetApp filer LUN mounted on a Linux host.

1. Backup the file system being stored on the NetApp filer LUN using operating system commands or a snapshot record. Note that fdisk is used when resizing the partition on the LUN. Anytime fdisk is used on a LUN, you should back it up first.

2. Resize the LUN on the filer:

```
data> lun resize /vol/linuxblocks/linuxblockslun0 20g
lun resize: resized to:   20.0g (21476206080)
```

3. You must now make the space available to the Linux host. The first step to accomplish this is to unmount the file system. On the iSCSI software initiator, the easiest way to accomplish this is to comment the mount point's entry in the /etc/fstab.iscsi file and the run "service iscsi restart." On either FCP or an iSCSI hardware initiator, a simple umount on the mount point will suffice.

4. You should now rescan the SCSI bus. Under the iSCSI software initiator, the restart of the iSCSI service that you performed in the last step will accomplish this. Under FCP, this would be:

```
[root@borg2of6 etc]# modprobe –r qla2300
[root@borg2of6 etc]# modprobe qla2300
```

   Note: Using "modprobe" in this way will unload and reload the Qlogic HBA driver. This requires that all I/O to any LUN accessed using this driver cease, and the data is briefly offline while this operation runs. A less intrusive approach to rescanning the SCSI bus can be found here. This script is available by default under SuSE Linux. On Red Hat Linux, it must be installed.

5. You resize the partition using fdisk. The "fdisk –l" command will show that the size of the disk has changed, but the partition has not:

```
[root@borg2of6 dev]# [root@borg2of6 /]# fdisk –l
   <some lines omitted>
Disk /dev/sdb: 64 heads, 32 sectors, 20480 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1     10240  10485744   83  Linux
```

   Note that the size of the disk "/dev/sdb" is shown as 20,480 cylinders, but the partition "/dev/sdb1" has only 10,240 cylinders. This needs to be fixed. We use fdisk to change the LUN itself, not the partition. The first partition on the disk is deleted, and a new partition of twice the size is created in its place. This does not affect the data on the disk. All the data in the file system remains intact. The following code listing illustrates this:

```
[root@borg2of6 /]# fdisk /dev/sdb
   <some lines omitted>
Command (m for help): d
Partition number (1-4): 1

Command (m for help): n
Command action
   e   extended
```

```
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-20480, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-20480, default 20480):
Using default value 20480

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

Calling ioctl() to re-read partition table.
   <some lines omitted>
Syncing disks.
```

6. You must now resize the file system. Although the partition is now the correct size, the file system is not. If you were to mount the partition in its current state, you would still not see any additional space. The following code listing illustrates this:

```
   <mount point in /etc/fstab.iscsi uncommented>
[root@borg2of6 /]# service iscsi restart
Stopping iSCSI: sync umount sync iscsid iscsi
Starting iSCSI: iscsi iscsid fsck/mount
   <some lines omitted>
[root@borg2of6 /]# df /lun0
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/iscsi/bus0/target0/lun0/part1
                      10321192     32832   9764076   1% /lun0
```

   Note that the “df” command reports the file system size as about 10GB, whereas the “fdisk –l” command is reporting the partition and disk as both being 20GB. To fix this you must run the “resize2fs” command. However, the “e2fsck –f” command is a prerequisite. The following code listing contains the full procedure:

```
[root@borg2of6 /]# e2fsck -f /dev/sdb1
e2fsck 1.26 (3-Feb-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 12/1310720 files (0.0% non-contiguous), 49346/2621436 blocks
[root@borg2of6 /]# resize2fs /dev/sdb1
resize2fs 1.26 (3-Feb-2002)
The filesystem on /dev/sdb1 is now 5242876 blocks long.
```

7. At this point, you can remount the file system. Under the iSCSI software initiator, this would require uncommenting the entry in the “/etc/fstab.iscsi” file and rerunning “service iscsi restart.”

Under FCP and an iSCSI hardware HBA, a simple mount of the file system's mount point works fine. Running "df" at this point shows that the file system size is now what we would expect:

```
[root@borg2of6 /]# df /lun0
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/iscsi/bus0/target0/lun0/part1
                      20642412     32832  19770720   1% /lun0
```

The following table summarizes the process of resizing a filer LUN mounted on a Linux host:

Operation	Purpose
1. Backup the file system being stored on the LUN	Ensure that you can protect the data in the event of any problems during the procedure. This is appropriate anytime fdisk is used on a LUN.
2. Resize the LUN from within the filer.	Makes additional space available to the LUN.
3. Unmount the LUN.	Allows the LUN's partition information to be changed on the Linux host.
4. Rescan the SCSI bus.	Allows the Linux host to see the additional space on the LUN.
5. Resize the LUN using fdisk.	Allows the Linux host partition to use the additional disk space within the partition.
6. Resize the file system.	Restores the data on the LUN to the state when the Snapshot record was created.
7. Mount the LUN.	All of the file system is now open for reading and writing by users. The data in the file system is intact, and the additional space is now available for use.

7. Importing a Partition Stored on a NetApp Filer LUN Under Linux

The following procedure will allow you to import a file system stored on a NetApp filer LUN onto a Linux host. The example assumes that you are using a backup copy of an existing LUN. This should be an extremely common operation. This allows you to access the files within a Snapshot record, so that file-by-file restore can be accomplished, thus effectively duplicating the .snapshot directory, available under NFS. It is also a common way of backing up a file system stored on a NetApp filer LUN. Another use for this procedure would be testing and verification of a file system on a NetApp filer LUN. The procedure is detailed in the following steps.

1. We begin with a 10GB LUN mounted on the Linux host as /mnt/datalun. This LUN is stored at /vol/linuxblocks/linuxblockslun0. It is LUN0 under the igroup linuxigroup.
2. First we follow the procedure to create a Snapshot record of the LUN. See Section 4.

```
[root@borg2of6 etc]# sync
[root@borg2of6 etc]#
```

```
data> snap create linuxblocks linuxblockssnap
data> snap list linuxblocks
Volume linuxblocks
working...

  %/used       %/total  date          name
```

```
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  Dec 02 15:48  linuxblockssnap

data>
```

3. Next, create a backup copy of the LUN, using the Snapshot record we just created:

```
data> lun create -b /vol/linuxblocks/.snapshot/linuxblockssnap/linuxblockslun0
/vol/linuxblocks/linuxblockslun0snap
data> lun map /vol/linuxblocks/linuxblockslun0snap linuxigroup
lun map: auto-assigned linuxigroup=1
data> lun show
        /vol/linuxblocks/linuxblockslun0         10g (10737418240)   (r/w,
online, mapped)
        /vol/linuxblocks/linuxblockslun0snap     10g (10737418240)   (r/w,
online, mapped)
```

4. Now rescan the SCSI bus to see the new LUN. See Section 5 for details on this step. The following example assumes iSCSI. FCP would be similar:

```
[root@borg2of6 etc]# service iscsi restart
Stopping iSCSI: sync umount sync iscsid iscsi
Starting iSCSI: iscsi iscsid fsck/mount
[root@borg2of6 etc]# /dev/iscsi/bus0/target0/lun0/part1: clean, 12/1310720
files, 49346/2621436 blocks

[root@borg2of6 etc]# fdisk –l
<some lines omitted>
Disk /dev/sdb: 64 heads, 32 sectors, 10240 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1     10240  10485744   83  Linux

Disk /dev/sdc: 64 heads, 32 sectors, 10240 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdc1             1     10240  10485744   83  Linux
[root@borg2of6 etc]#
```

5. Note that a new LUN has appeared, /dev/sdc, and this LUN has a partition, /dev/sdc1. We now simply mount this partition, using the syntax preferred by the Linux iSCSI software initiator. See the README file that accompanies the Linux iSCSI software initiator on the details of this syntax. We need to add a mount point to the /etc/fstab.iscsi file as follows:

```
/dev/iscsi/bus0/target0/lun1/part1 /mnt/datalunsnap ext3 defaults 0 0
```

   Under FCP, we would simply add a mount point in /etc/fstab and mount it like any normal file system.

6. At this point, create the mount point and mount it. The following assumes iSCSI:

```
[root@borg2of6 etc]# mkdir /mnt/datalunsnap
[root@borg2of6 etc]# service iscsi restart
Stopping iSCSI: sync umount sync iscsid iscsi
Starting iSCSI: iscsi iscsid fsck/mount
[root@borg2of6 etc]# /dev/iscsi/bus0/target0/lun0/part1: clean, 12/1310720
files, 49346/2621436 blocks
/dev/iscsi/bus0/target0/lun1/part1: recovering journal
/dev/iscsi/bus0/target0/lun1/part1: clean, 12/1310720 files, 49346/2621436
blocks

[root@borg2of6 etc]#
```

   Under FCP, we would simply mount the file system using the mount command.

The procedure is now complete. The backup copy of the LUN has been imported onto the Linux host and is available at /mnt/datalunsnap. The data in this LUN is consistent with the state of the source LUN at the time the Snapshot record was created.

The same procedure can be used to import a LUN from a different host. The limitations are that the file system and OS versions must match in order for this procedure to succeed.

The following table summarizes the process of importing a filer LUN on a Linux host:

Operation	Purpose
1. Make the LUN to be imported accessible to the Linux host.	The LUN must be visible to the Linux host in order to be imported. On iSCSI, this means the filer must be connected to an IP network that the Linux host can see. On FCP, the Linux host must be on a common FCP network with the filer. Either way, the LUN must be in an igroup that contains the Linux host.
2. Rescan the SCSI bus.	Allows the Linux host to see the LUN.
3. Mount the LUN on the Linux host.	Allows the Linux host to use the space on the LUN as well as access any existing data on the LUN.

8. Conclusions

Once available for Linux, SnapDrive will be the preferred Network Appliance method to accomplish the objectives outlined in this technical report. Until then, most of the features provided by SnapDrive can be accomplished by other means. This technical report details these procedures for the customer who wishes to use these features until SnapDrive becomes available on Linux.

NetApp®

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089
www.netapp.com