Rational ClearCase Backup and Restore Using a NetApp Filer in a UNIX® SAN Environment

Maneesh Jain | July 2003 | TR 3269

TECHNICAL REPORT

Network Appliance Inc.

Table of Contents

1) Purpose

This document covers backup and recovery of ClearCase data running in an interop environment, with the VOB database and storage pools on a Network Appliance™ filer. It focuses on the backup and restore procedures for ClearCase data when the filer has been configured for a SAN.

Information on how to configure the filer in a SAN environment can be found on NOW™ (NetApp on the Web, NetApp's online support site, open to all current NetApp customers) or in our documentation (*HBA Installation and Setup Guide*) that comes with the filer.

2) Key Benefits

The performance and reliability of the backup and recovery operations are critical to effective ClearCase operation. Network Appliance provides unique functionality that affords the ClearCase administrator significant reliability and performance benefits in both backup and recovery. The key feature of the Network Appliance approach to backup is the use of Snapshot™ technology. For more information on how Snapshot works, see File System Design for an NFS File Server Appliance by Dave Hitz and James Lau.

Snapshots are critical because they allow the ClearCase administrator to quickly and easily create an image of the entire filer volume including the LUNs containing file systems that hold VOB and View storage. After the Snapshot is taken, the VOB can be returned to normal operation. A backup can then be written to tape or to a NearStore™ appliance utilizing SnapVault™ while ClearCase is being used. NearStore is a disk-based online storage solution that uses the existing Data ONTAP™ operating system and WAFL® file system with ATA disk drives. For more information about NearStore refer to the following: http://www.netapp.com/products/nearstore/

A client-free backup scheme can also be implemented by mounting the new LUN directly on the backup host. Actual backup-to-tape performance is of secondary importance, since the impact of this operation on ClearCase's performance is negligible. For the backup operation, the paramount issue is that the VOB database must be locked and unlocked in a minimal amount of time.

VOB databases cannot be locked for the hours that may be required to perform an offline backup; however, locking the database for only a few seconds may be acceptable. Snapshot functionality affords the ClearCase administrator the ability to lock the VOB database for only a few seconds, during which a Snapshot is taken, and then to unlock the VOB database for continued normal operation. One can also take Snapshot copies several times per day and back up only one of them to tape. Effectively, you can take an online backup of your entire VOB every few hours. (*Note*: The amount of storage overhead associated with a Snapshot is based upon the number of blocks that are different between the Snapshot and the active file system. Using LUN storage requires space reservation to be enabled within the volume to ensure that the host will never see a volume full induced write error. As a result, the initial Snapshot requires 1X the amount of disk space in use by the LUNs in the volume. Subsequent Snapshots require incremental block allocation to cover the data blocks that have changed. As this increases, the size of the Snapshotstorage requirements will increase as well. Thus, recent Snapshot images are generally less costly to store than older ones.)

Snapshots also provide advantages for the restore operation. If the data that you need is in a Snapshot, you can restore the entire VOB, regardless of size, in a few minutes. This is accomplished with the Data ONTAP SnapRestore® command. If your data protection architecture includes SnapVault, and if your data is not in an online Snapshot, it might be restorable from the NearStore device. If you save several days' worth of Snapshots, the chances are that you will never need to restore from tape at all, barring a failure on the filer itself. (*Note*: NetApp always recommends that all data be backed up to tape often. Snapshots should be used as a supplement, not a replacement, for tape backup.)

3) Assumptions

It is assumed that you are familiar with *Rational ClearCase: Using with NetApp Filers*, as well as *Backup and Restore of ClearCase Data*.

If the filer is configured for NAS only, then follow the above two links for help. However, since the FAS960 can be deployed both in a NAS as well as SAN configuration, this document will cover backup and restore only in a SAN configuration.

General familiarity with SANs, as well as LUNs, is expected.

The name of the filer is `hansel`, a FAS960 running Data ONTAP 6.4R1
The name of the VOB server is `ccsvr`, a Sun™ Solaris™ 8 machine, running ClearCase version 2002.05.00
The VOB server is configured to perform remote shell operations on the filer
The filer `hansel` is resolvable by name on the VOB server
The volume `hansel:/vol/vol2` contains a qtree called qt0
The qtree `hansel:/vol/vol2/qt0` contains a LUN called `lun0` and is mounted at /halun0 on `ccsvr`
The qtree hansel:/vol/vol2/qt0 contains a LUN called `lun1` and is mounted at `/halun1` on `ccsvr`
The following entries exist in the `/etc/vfstab` file on `ccsvr`:

```
/dev/dsk/c4t18d0s6   /dev/rdsk/c4t18d0s6   /halun0  ufs   3   yes   logging
/dev/dsk/c4t18d1s6   /dev/rdsk/c4t18d1s6   /halun1  ufs   3   yes   logging
```

The VOB tags are `/vobs/ebc1` and `/vobs/ebc2`
Root access is required on `ccsvr`
SnapRestore must be licensed on the filer

The space_reservations option must be enabled on any volume that contains a LUN. This feature guarantees that there will always be enough free space to perform a write operation to the LUN in case a Snapshot has been taken and 100% of the blocks have changed. With NFS, a file system-full error is generally tolerable at the host level since the filer and WAFL own and manage the file system. With FCP and block devices, however, it is up to the host to manage the file system, and therefore the host will always assume that 100% of disk is available to be overwritten. In the event that the filer volume is full, the host attempts to write to a block, and WAFL is unable to provide a free block, the filer will return a SCSI write error to the host. This can be fatal for the host depending on what data is stored in the file system and what type of file system was created on the host. Enabling space_reservations will prevent this scenario.

4) Backup

Taking a backup involves the following steps:

Perform some housecleaning to keep a number of Snapshots online (in this example, four are being kept online)

Lock the VOB

Flush and suspend I/O to the file system

Take a Snapshot copy of the file system

Unlock the file system

Unlock the VOB

These steps can be easily scripted as follows (`dobackup.sh` shell script):

```
#!/bin/sh

# Rename and delete old snapshots
rsh hansel snap delete vol2 cc_old3
rsh hansel snap rename vol2 cc_old2 cc_old3
rsh hansel snap rename vol2 cc_old1 cc_old2
rsh hansel snap rename vol2 cc cc_old1

# Lock the VOB
cleartool lock vob:/vobs/ebc1

# Flush and suspend write I/O
lockfs -w -c "locked for snapshot" /halun0
lockfs -f /halun0

# Take a new snapshot
rsh hansel snap create vol2 cc

# Resume normal I/O
lockfs -u /halun0

# Unlock the VOB
cleartool unlock vob:/vobs/ebc1

# Perform your backup using dump or other operating system commands.
# For example:
# rsh hansel dump 0ufbln rst0a 63 /vol/vol2/.snapshot/cc/
```

The options for dumping data from a filer to tape are beyond the scope of this technical report. They are thoroughly documented in the *NetApp Data ONTAP System Administration Guide* http://now.netapp.com/NOW/knowledge/docs/ontap/rel64r1/html/ontap/dpg/index.htm (*Note*: This requires a NOW account).

4.1. Integration with NearStore

After securing a Snapshot within the filer, there are many optional ways to archive the data. Traditionally, a dump-to-tape process would be employed to create a long-term archival or offline disaster recovery copy. Another alternative is to synchronize a copy to a second filer using SnapMirror® software. A new, lower-cost option is to use SnapVault software to copy the data to a Network

Appliance NearStore appliance.

There are many advantages to using a NearStore device. After the backup has been completed via SnapVault, a backup to tape can be made independent of the host and without additional load on the primary filer. Since SnapVault only transfers changed blocks, the backup process normally completes in a reduced time frame compared to direct backup to tape. With NearStore, the window for tape backups is much larger while maintaining filer performance and availability for the application. Also, NearStore can be used to consolidate backup storage and centralize all backups from many filers. Perhaps the most compelling advantage is that rapid data recovery is possible since the data files can be brought back online without having to first restore from tape.

For a discussion of filer and NearStore integration, see *SnapMirror and SnapVault Capacity Planning for the NearStore* . For more information about NearStore architecture, see *Nearline Applications for the Enterprise Utilizing NearStore*.

5) Restore

The example concerning restore assumes that the data required is still stored in Snapshot. If that is not the case, you will need to perform either a SnapVault restore or a restore-from-tape operation. The requirements to do this are comparable to a similar operation on a UNIX® system. The steps are beyond the scope of this paper, but are amply documented in the *NetApp Data ONTAP System Administration Guide*, which is available from NOW.

5.1 Full Volume SnapRestore

The following will restore the entire volume to a previous point-in-time copy from the **cc** Snapshot. In order to restore a whole volume, all LUNs on that volume have to be I/O flushed. Since in this example there is only one LUN (`lun0`) in the volume you will not end up with inconsistent LUNs.

Restoring the VOB storage area requires the following commands to be issued from `ccsvr`:

Lock the VOB

```
ccsrv# cleartool lock vob:/vobs/ebc1
```

Unmount the file system

```
ccsvr# umount /halun0
```

Restore the file system from the last Snapshot copy taken

```
ccsvr# rsh hansel snap restore -t vol -s cc vol2
WARNING! This will revert the volume to a previous snapshot.
All modifications to the volume after the snapshot will be
irrevocably lost.
Volume vol2 will be made restricted briefly before coming back online.
Are you sure you want to do this? yes
You have selected volume vol2, snapshot cc
Proceed with revert? yes
Volume vol2: revert successful.
```

Mount the file system

```
ccsvr# mount /halun0
```

Unlock the VOB

```
ccsvr# cleartool unlock vob:/vobs/ebc1
```

5.2 Full LUN SnapRestore

Since Snapshot occursat the volume or qtree level, the entire contents are captured as a unit each time a Snapshot copy is taken. If there are multiple LUNs or other multiprotocol files stored within that same volume, the single-file SnapRestore feature will allow the user to restore a subset of the volume while leaving the remainder of the files intact.

Although the previous example utilized a single LUN in a volume, there is no restriction in the number of LUNs that a volume may hold. A possible scenario leading to multiple LUNs in a volume could be the case where there are multiple VOBs, each having its own LUN.

Note: Please follow the *HBA Installation and Setup Guide* for detailed information on how to create a LUN on the filer or follow this link for information: http://now.netapp.com/NOW/knowledge/docs/ontap/rel64r1/html/ontap/san_sag/index.htm (requires NOW account).

Assume another LUN has been created as follows:

```
ccsvr# mkdir /halun1
ccsvr# rsh hansel lun create -s 10g -t solaris /vol/vol2/qt0/lun1
ccsvr# rsh hansel lun map /vol/vol2/qt0/lun1 ccsvr 1
ccsvr# devfsadm
ccsvr# format c4t18d1
[ ... output abbreviated ... ]
ccsvr# newfs /dev/rdsk/c4t18d1s6
[ ... output abbreviated ... ]
[ ... update /etc/vfstab ... ]

ccsvr# mount /halun1
```

Another VOB (ebc2) has been created with storage on `/halun1`. Backups would still be performed by running the `dobackup.sh` script with the appropriate changes to include **/**`halun1`.

```
ccsvr# ./dobackup.sh
```

To simulate a failure in the LUN, delete the source pool directory from the `ebc2` VOB.

ccsvr# \rm –r /halun1/ebc2.vbs/s

Note that any number of errors or data corruptions could result in the necessity to restore a data file. For the purposes of this example, we will restore the LUN from a previously saved Snapshot that contains the data file of interest.

The following commands will lock the VOB, unmount the existing file system, restore the single LUN from the Snapshot image (single-file SnapRestore), mount the LUN to `/ccvob`, and unlock the VOB:

```
ccsrv# cleartool lock vob:/vobs/ebc2
ccsvr# umount /halun1
ccsvr# rsh hansel lun offline /vol/vol2/qt0/lun1
ccsvr# rsh hansel snap restore -t file -s cc /vol/vol2/qt0/lun1

WARNING! This will restore a file from a snapshot into the active
file system. If the file already exists in the active file system,
it will be overwritten with the contents from the snapshot.
Are you sure you want to do this? y
You have selected file /vol/vol2/qt0/lun1, snapshot cc
Proceed with restore? y
ccsvr# rsh hansel lun online /vol/vol2/qt0/lun1
ccsvr# mount /halun1
ccsvr# cleartool unlock vob:/vobs/ebc2
```

This will have effectively restored only the ebc2 VOB, leaving the ebc1 VOB intact since that one is on a different LUN.

5.3 File-Level Access within a Snapshot Copy

The previous example depicts the scenario where an entire LUN in a volume containing multiple LUNs has been restored. The Snapshot capability allows the entire LUN to be replaced with a previous Snapshot in a matter of seconds through the simple manipulation of a set of pointers.

It is also possible to create a new LUN from Snapshot by utilizing the Snapshot as a backing store. If the new LUN is then mounted on the host system, single files from within the file system on the LUN can be copied (or restored) using standard UNIX commands.

This technique may also be employed to create a system where a second host obtains access to the Snapshot data. By simply performing the commands shown below on the second host, it is possible to access the Snapshot copies of the file systems on another host. The possible uses for this technology are limitless, and further discussion is beyond the scope of this technical report. However, as a primer, consider the following examples for which this technique may be useful:

Clientless backups performed from a second UNIX host

Creating a backup of a production environment for lab/development access

Disaster recovery scenarios

The steps in this section outline the process for creating a LUN with Snapshot as backing store. We assume that a Snapshot exists from previous execution of either the `dobackup.sh` script or from a manual process. There must also be sufficient disk devices present in the `/kernel/drv/sd.conf` to support access to the newly created LUN.

First, create the new LUN on the filer from the Snapshot and map it to the Solaris™ host:

```
ccsvr# rsh hansel lun create -b /vol/vol2/.snapshot/cc/qt0/lun1
```

```
/vol/vol2/qt0/templun
ccsvr# rsh hansel lun map /vol/vol2/qt0/templun ccsvr 2
```

Now mount the new LUN on the host to a temporary location:

```
ccsvr# mkdir /mnt/templun
ccsvr# devfsadm
ccsvr# mount /dev/dsk/c4t18d2s6 /mnt/templun
ccsvr# cd /mnt/templun
```

You can now access files within the `/mnt/templun` directory using standard UNIX commands.

It is advisable to unmount and destroy the LUN when access is no longer needed, as there is a limit to the number of Snapshots that can exist in a volume at one time. Tying up Snapshot for nonproduction LUNs on a continuous basis may impact the ability of a scheduled process (e.g., a Snapshot cron job) to execute normally. In the case where long-term access to a Snapshot copy is required, it is best to perform a single-file `snap restore` command to generate a new LUN in the volume and then mount that LUN on the host instead of the Snapshot copy.

6) Conclusions

A Network Appliance filer offers compelling advantages in terms of backup and recovery. Use of Snapshots, combined with conventional backup-to-tape techniques, can dramatically optimize the ClearCase backup operation. Retaining a number of online Snapshots allows for restores without the necessity to restore from tape in many circumstances. Backup and recovery performance is dramatically improved over conventional local disk configurations.

7) References

Information in this paper was taken from the following technical reports:

Oracle9*i*™ for UNIX: Backup and Recovery Using a NetApp Filer in a SAN Environment
Rational ClearCase: Using with NetApp Filers
Backup and Restore of ClearCase Data

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