NETAPP TECHNICAL REPORT

Using SnapVault with SnapDrive for UNIX

Anil Degwekar, NetApp
TR-3641

Abstract

This technical report explains how to use SnapDrive® for UNIX® along with SnapVault®. This information is intended for users of NetApp™ storage systems that utilize these products.

SnapDrive for UNIX does not include dedicated commands to control SnapVault operations. However, users can operate these two products together by writing scripts or by following a few manual steps, which are explained in this report.

You should be familiar with the workings of Data ONTAP®, SnapDrive for UNIX, and SnapVault before using the information in this document.

.

Table of Contents

1 Introduction

1.1 About SnapDrive for UNIX

SnapDrive for UNIX is a tool for storage provisioning and Snapshot™ copy management on UNIX systems that connect to NetApp FAS series storage systems. It provides UNIX system administrators easy access to storage provisioning functionality such as creating LUNs, connecting them to the UNIX system, creating a volume group and file system on these LUNs, and expanding the volume groups by using simple commands on the UNIX console. It also allows the creation of copies of user data that are host file-system-consistent, as well as the restoration of data from these Snapshot copies. Other NetApp software products, such as SnapManager® for Oracle® and SnapManager for SAP®, use the underlying functionality provided by SnapDrive for UNIX to enable Snapshot copy management, which is integrated into applications that use this storage.

1.2 About the data protection features of Data ONTAP

Data ONTAP has several unique features for data protection, including SnapMirror® and SnapVault. With SnapMirror you can schedule regular, automatic replication of file system Snapshot copies of a volume or qtree onto another volume or qtree (usually on a different storage system). SnapVault protects the data in one or more qtrees in a series of Snapshot copies stored on a separate storage system. SnapVault maintains an online, asynchronous, permanent read-only replica of the qtree data.

For information on using SnapDrive for UNIX with SnapMirror, refer to Technical Report 3611. The current technical report deals mainly with using SnapDrive for UNIX with SnapVault.

1.3 Interoperability between SnapDrive for UNIX and SnapVault

It is natural for users of storage to use both of these products together, first to use SnapDrive for UNIX to create host-consistent Snapshot copies, and then to use SnapVault to back up these Snapshot copies on a separate storage system.

Although there is no direct integration between these two product groups (that is, SnapDrive for UNIX does not provide any specific commands or CLIs to control the way SnapVault behaves), customers can use these products together. This technical report explains the various ways in which customers can use SnapDrive for UNIX with SnapVault. In addition to explaining the steps that users must take to ensure correct use of these products, it gives sample deployment scenarios. This information should be used in conjunction with the user guides published for each of these products. This technical report does not reproduce all of the information in the product user guides.

2 SnapDrive for UNIX usage with SnapVault

2.1 Example data protection deployment scenario

In the example configuration shown in Figure 1, a UNIX host (alpha) is connected to a primary NetApp storage system (toaster), and the data on toaster is backed up to a secondary storage system (oven) for data protection. The secondary storage system may be at a nearby location or at a distant location. The UNIX host uses LUNs created on qtrees within the primary storage system. The UNIX host alpha is connected to toaster via an FCP fabric. The connection between alpha and oven is over IP WAN. In normal usage, the UNIX system alpha does not use any storage from oven. However, it may have connectivity to oven over IP SAN (iSCSI), and that connectivity must be tested before the deployment of this configuration. This connectivity can be

used for recovery of data.[1] Note that recovery of data is possible even without this connectivity, as explained later in this document.

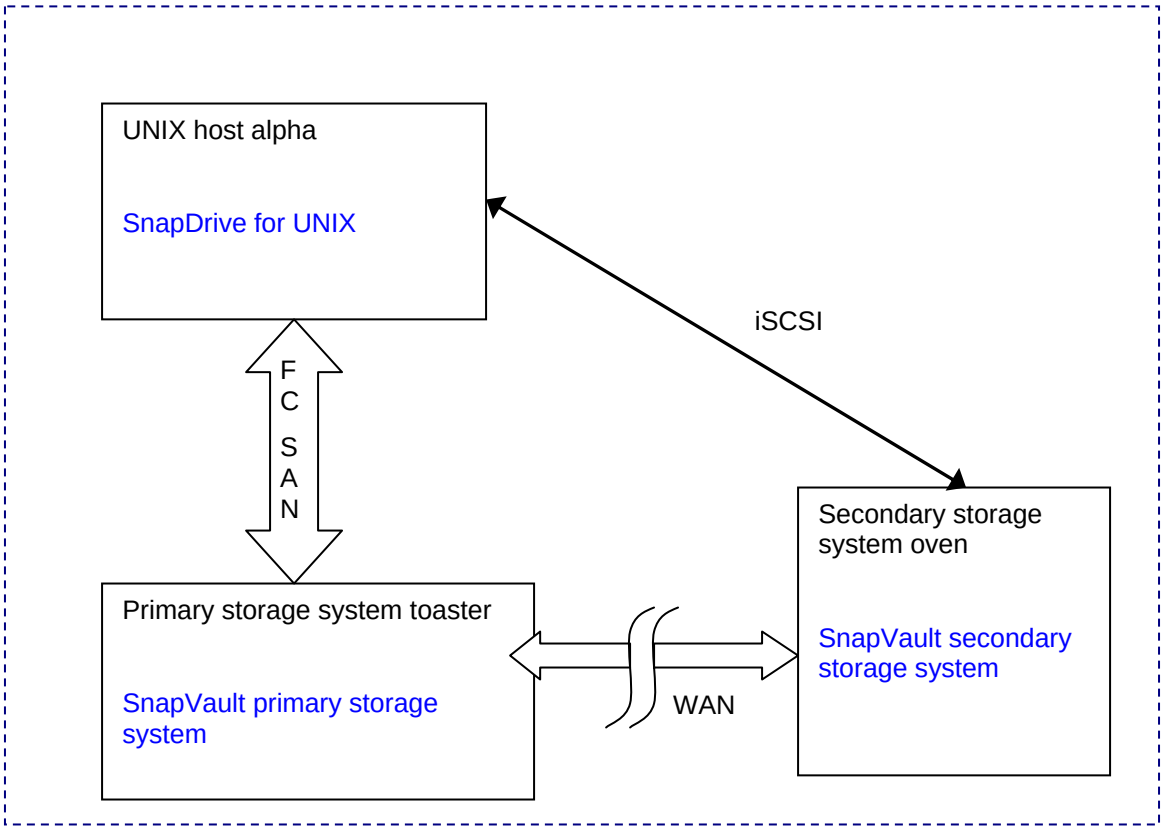

Figure 1) SnapDrive for UNIX with SnapVault configuration 1.

This configuration can be further enhanced by using host clustering technology on the host side, which handles failures of the host node.

2.2 Software installation and setup

This configuration can be installed and set up as explained in the installation guides supplied with the storage systems. Each storage system is configured with the required licenses (including licenses required for SnapDrive for UNIX and those required for SnapVault). No specific order is required for installation. You can test the connectivity between the systems as described in the following sections.

2.2.1 Connectivity between the UNIX host and the primary storage system

Verify that you can create LUNs on qtrees on the primary storage system by using SnapDrive for UNIX. To do this, follow the steps described in the *SnapDrive for UNIX Installation and Administration Guide.* You must select fcp as the protocol of choice in the SnapDrive configuration file (snapdrive.conf) on alpha.[2] Perform the following steps to verify that all aspects of the connection work properly:

---

[1] The IP SAN connectivity may not give the same performance as the FCP SAN connectivity with the primary storage; however, it allows the host alpha to continue to operate and access its data until the primary is brought up again. Also, there may be platform-specific restrictions that do not allow simultaneous use of FCP and iSCSI connectivity on the same host.

[2] It is also possible to use iSCSI connectivity between alpha and toaster.

On the storage system:

1. Create a volume vol1 on toaster.

On the UNIX host:

2. Create a test LUN and mount it to the host by using this command:
   `snapdrive storage create –fs /mnttest –lun toaster:/vol/vol1/qtree1/testlun –lunsize 1g`
3 Copy some data to the LUN by using a command such as:
   `cp –r /usr/bin/* /mnttest`
4 Create a Snapshot copy of this data by using a command such as:
   `snapdrive snap create –fs /mnttest –snapname trial`
5 Delete the data that you just copied:
   `rm –r /mnttest/*`
6 Restore the data by using a command such as:
   `snapdrive snap restore –fs /mnttest –snapname trial`
7 Verify that the data was properly restored by using a command such as:
   `ls –l /mnttest`

Once connectivity has been established, you can delete this Snapshot copy by using a command such as `snapdrive snap delete –snapname toaster:/vol/vol1:trial`. You can then delete the test LUN by using a command such as `snapdrive storage delete –fs /mnttest`.

2.2.2 Connectivity between SnapDrive for UNIX and the secondary storage system

If there is connectivity between alpha and oven, you must test this connectivity during installation. Verify that you can create LUNs on the secondary storage system by using SnapDrive for UNIX on the UNIX host. To do this, follow the steps described in the *SnapDrive for UNIX Installation and Administration Guide*. You must select iscsi as the protocol of choice in the SnapDrive configuration file (snapdrive.conf) on alpha.[3] Perform the same steps that you followed in the previous example, substituting the storage system name oven in place of toaster. Note that even though the connectivity is different here (iSCSI instead of FCP), the command syntax remains the same. At the end of the test, change the preferred connectivity in the snapdrive.conf file back to FCP.

2.2.3 SnapVault installation and setup

Set up a SnapVault relationship between the two storage systems. On the primary storage system, follow these steps to configure and enable SnapVault:

```
toaster> ndmpd on
toaster> options snapvault.enable on
toaster> options snapvault.access host=oven
```

On the secondary storage system, follow these steps to configure and enable SnapVault:

```
oven> ndmpd on
oven> options snapvault.enable on
oven> options snapvault.access host=toaster
oven> snapvault start –S toaster:/vol/vol1/qtree1 oven:/vol/vol2/alpha
```

[3] In the sample configuration, the connectivity between alpha and oven is iSCSI, but in theory FCP connectivity also works in the same way. Most installations use iSCSI if the secondary system is at a remote location.

The last command starts the baseline transfer of data. The qtree on the primary storage system (`toaster:/vol/vol1/qtree1`) should be present. The qtree on the secondary storage system should not exist; SnapVault creates it automatically. However, the volume (`oven:/vol/vol2`) should exist.

Important considerations about SnapVault configuration:

- Qtree is the basic unit of SnapVault backup and restore. Therefore the SnapVault relationship shown in the previous example is from a qtree on the primary storage system to another qtree on the secondary storage system. You can use SnapVault to back up non-qtree data also. However, in this technical report the examples illustrate backing up qtree data only.
- Primary storage qtrees from multiple storage systems can all be backed up to associated qtrees on a single SnapVault secondary volume. Use this configuration only if you plan to take a single Snapshot copy of all these primary storage qtrees (for example, when the data is part of a single volume group on the host).
- The versions of Data ONTAP used between the two systems should be compatible for using SnapVault.
- Space reservation should be enabled for the LUNs on the secondary system volume. Enabling space reservation ensures that there is always space for writes to the LUN during SnapVault transfers.
- Do not set up the backup schedule by using the SnapVault scheduling mechanism at this stage. Refer to Section 3, “Backing Up Data.”
- A common variation of the basic SnapVault backup deployment adds a tape backup of the SnapVault secondary storage system. This deployment is also possible with SnapDrive for UNIX. Likewise, other best practices mentioned in the SnapVault Best Practices Guide are also applicable.

2.3 Provisioning considerations

Once the setup has been configured as just described, further use of SnapDrive for UNIX on the UNIX host is straightforward. When provisioning, use the qtree that is configured for backup.

Example command for storage provisioning:

```
snapdrive storage create –fs /mnt/data –hostvol SduDg/SduHv –lun
toaster:/vol/vol1/qtree1/lun1 –lunsize 100g
```

3 Backing up data

To back up data by using SnapVault, you must make Snapshot copies of data on the primary storage system. There are two possibilities for making these copies: by using the SnapDrive for UNIX command line or by using a script running on the host.

Note: You should not use the Snapshot copy scheduler that is built in to Data ONTAP or SnapVault. The Snapshot copy process should be initiated by SnapDrive for UNIX, to ensure that the Snapshot copies are host-file-system consistent.

3.1 Backing up unscheduled Snapshot copies

You can use the SnapDrive for UNIX command line to make Snapshot copies of data at unscheduled intervals.

Example SnapDrive for UNIX command for making a Snapshot copy:

```
snapdrive snap create –fs /mnt/data –snapname 28sep649pm
```

To back up this Snapshot copy to the secondary storage system, you can use the following command on the SnapVault secondary storage system:

```
oven> snapvault update –s 28sep649pm /vol/vol2/alpha
```

Wait for the transfer to complete[4]. Check the status of the transfer by using this command:

```
oven> snapvault status /vol/vol2/alpha
```

When the transfer is complete, make a Snapshot copy of the data by using a command such as:

```
oven> snap create vol2 28sep649pm
```

Use the same name for the Snapshot copy on both systems, to make it easy to keep track of the Snapshot copy names[5].

3.2 Scheduling regular Snapshot copies by using a UNIX scheduler

All UNIX systems have the built-in cron scheduler, which can be used to schedule regular jobs. Following is an example of a script that schedules Snapshot copies to occur at hourly, nightly, and weekly intervals.

```
#!/bin/sh
# sdu-fs-snapshot.sh
# This sh script makes hourly/nightly/weekly Snapshot copies using SDU
# First parameter -- filesystem mount point path
# Second parameter -- storage system volume path
# Third parameter -- name of the Snapshot copy
# Fourth parameter -- number of Snapshot copies to keep
#
# Usage: sdu-fs-snapshot.sh <mntpoint> <storage-volume> snapname> <number>
# You can also add this to the crontab with the appropriate schedule
tmpname=sdu-fs-snapshot
if [ $# -ne 4 ]
then
  echo "Usage: $0 <mntpoint> <storage-volume> <snapname> <number>"
  exit 1
fi
RET=0
/usr/sbin/snapdrive snap create -fs $1 -snapname $tmpname -force -noprompt
RET=$?;
if [ $RET -ne 0 ]; then
  time=’date’
  echo "$0: SDU Snapshot creation failed on filespec $1 at $time"
  exit 1
```

[4] If you add the -w option to this command, it waits until the update is complete. This can be useful if you want to invoke this command from within a script.

```
fi
#
# Now loop through the older Snapshot copies, starting from the oldest
#
# First, delete the oldest Snapshot copy
/usr/sbin/snapdrive snap delete ${2}:${3}.$4
# Now, increment the subscript for all previous Snapshot copies
n=$4
while [ $n -gt 0 ]
 do
   m=`expr $n - 1`
   /usr/sbin/snapdrive snap rename ${2}:${3}.$m ${3}.$n
   n=$m
 done
# Now, rename the previous Snapshot copy to <snapname>.0
/usr/sbin/snapdrive snap rename ${2}:$tmpname ${3}.0
# done, exit
exit 0
```

The script takes four parameters—the file system mount path on the host; the path of the primary storage system volume; the Snapshot copy name; and the number of copies to keep. Each Snapshot copy name is appended with a number subscript, starting from 0 (0 is the latest). This scheme fits well with the Snapshot naming scheme that is used by SnapVault. Using such a naming scheme, you can transfer these copies to the SnapVault secondary storage system automatically.

To add this script to the crontab, add these entries in the crontab file:

```
0 1-23 * * * /sdu-fs-snapshot.sh /mnt/data toaster:/vol/vol1 sv_hourly 10
0 0 * * 0,1,2,3,4,5 /sdu-fs-snapshot.sh /mnt/data toaster:/vol/vol1 sv_nightly
12
0 0 * * 6 /sdu-fs-snapshot.sh /mnt/data toaster:/vol/vol1 sv_weekly 20
```

This example creates three types of Snapshot copies—hourly, nightly, and weekly. The hourly copies (named `sv_hourly.x`) are made every hour except midnight. Ten copies are retained on the primary storage system (named `sv_hourly.0` through `sv_hourly.9`). Similarly, nightly Snapshot copies (named `sv_nightly.x`) are made at midnight every day of the week except Saturday. At midnight on Saturday, a weekly copy (named `sv_weekly.x`) is made.

To back up these copies to the secondary storage system, use these commands:

```
oven> snapvault snap sched –x vol2 sv_hourly 26@mon-sun@1-23
oven> snapvault snap sched –x vol2 sv_nightly 36@mon-fri,sun@0
oven> snapvault snap sched –x vol2 sv_weekly 52@sat@0
```

[5] This is a matter of convenience, and is not mandatory.

This example creates similar copies on the secondary, and follows the same schedule as the primary. It retains 26 hourly Snapshot copies, 36 nightly Snapshot copies, and 52 weekly Snapshot copies on the secondary storage system.

Important considerations about scheduled backup:

- The time of day on all three systems (alpha, toaster, and oven) should be synchronized, so that the schedules match well. If the systems are in different time zones, the schedule must be adjusted accordingly.
- The Snapshot copy names should be identical, because the SnapVault secondary storage system looks for these names during the transfer.
- Snapshot copies that are scheduled on the secondary storage system with the `snapvault snap sched -x` command are created 5 minutes after the hour that you specify. This delay is usually sufficient to give the script that is running on the UNIX host enough time to create Snapshot copies before the secondary storage system is updated from them.
- In this example, more Snapshot copies are retained on the secondary system, because typically it has a larger storage space.
- In case of failures in making Snapshot copies on the host, cron sends an e-mail message to the UNIX user account.
- SnapVault transfer status on the secondary storage system can be monitored by using the `snapvault status` command.
- Snapshot copies on the secondary system can be further protected by a SnapMirror backup or tape backup mechanism.

4 Restoring data

You may need to restore data from the backup copies in case you lose data because of a manual error or because of a disaster. You may also want to access an older copy of the data for reference.

There are three possible ways in which data can be restored from the secondary storage system: by directly accessing it from the UNIX host; by restoring it in a new qtree on the primary storage system; and by restoring it in place. The following sections describe how each of these methods works.

In this example, the restore is made from a weekly backup that was made 40 weeks ago, so the Snapshot copy name is `sv_weekly.40`. Because this exceeds the retention period on the primary storage system (which keeps only 20 weekly backups in the example), the data must be restored from the secondary storage system.

4.1 Accessing data from the secondary storage system

If there is connectivity between the UNIX host and the secondary storage system, the UNIX host can directly access data from the secondary storage system. This is not considered a normal restore, but it is essentially a connection to the secondary system's copy of the data. This is the fastest way to restore data, but the performance of this connection may not be good. It can be used to restore a small subset of the data.

There is no direct command in SnapDrive for UNIX to accomplish this. However, there is a four-step procedure that you can follow to achieve the same objective.

Step 1: List the Snapshot copies on the secondary storage system.

Use a SnapDrive for UNIX command such as:

```
snapdrive snap list –filervol oven:/vol/vol2
```

The Snapshot copy of interest (`sv_weekly.40` in this case) should be listed in the output list of Snapshot copies. It is listed as a "non-snapdrive snapshot," because SnapDrive for UNIX is not aware of the SnapVault data migration. However, the following steps enable you to access the data in this Snapshot copy.

Step 2: Create a FlexClone® volume based on the Snapshot copy of interest.

Use FilerView® or the storage system CLI to create a FlexClone volume from the SnapVault secondary system[6]. This FlexClone volume should be based on the Snapshot copy of interest (`sv_weekly.40` in this case).

In FilerView, you can use the Create FlexClone wizard. The CLI to create a FlexClone volume on the storage system console is:

```
oven> vol clone create vol40w –s volume –b vol2 sv_weekly.40
```

Here vol40w is the new FlexVol® clone, which is based on the volume vol2 and its Snapshot copy, sv_weekly.40.

If the UNIX host is configured for remote shell access to the storage system, then this command can be executed from the host by using rsh/ssh syntax. Otherwise, it must be executed from the storage system console.

Step 3: Make the LUN in the FlexClone volume online.

The new FlexClone volume, vol40w, contains a copy of the LUN lun1. In FilerView, use the LUN Manage menu to make this LUN online. Also, verify that the LUN is unmapped. The following step maps it to the host.

Step 4: Connect to the Snapshot copy.

On the UNIX host alpha, use a SnapDrive for UNIX command such as:

```
snapdrive storage connect –fs /mnt/remote –hostvol SduDg/SduHv –lun
oven:/vol/vol40w/alpha/lun1
```

This should create a new file system at the new mount point (/mnt/remote) with the contents of the Snapshot copy taken from the original data. Note that in this command you must specify a new mount point (/mnt/remote in this case). But the disk group name (SduDg in this case) and the host volume name (SduHv in this case) must be identical to the ones that you specified when creating the storage. The full LUN path from the FlexClone volume must also be provided.

If the original LUN (/mnt/data in this case) is still connected to the host, use a different disk group name in the previous command. Some volume managers may not allow you to rename the disk group name. In that case, you must disconnect the original LUN before accessing the Snapshot copy.

If the original data was created with the `nolvm` option (to create the file system directly on a raw LUN), then the same option must be provided at the time of storage connect.

You should use this connection only temporarily. While the FlexClone is present, the backing Snapshot copy is locked and cannot be deleted. This interferes with the SnapVault Snapshot copy retention mechanism. When your work is finished, unmap the LUN and destroy the FlexClone volume on the secondary system.

Step 5: Disconnect from the Snapshot copy.

On the UNIX host alpha, use a SnapDrive for UNIX command such as:

```
snapdrive storage disconnect –fs /mnt/remote
```

---

[6] The FlexClone license should be enabled on the secondary system.

This step unmaps the LUN and disconnects it from the host.

Step 6: Destroy the FlexClone volume.

In FilerView, you can use the Manage Volumes menu to destroy the FlexClone volume. The CLI commands to make the volume offline and then destroy it are:

```
oven> vol offline vol40w
oven> vol destroy vol40w
```

4.2 Restoring data to a new qtree on the primary storage system

Use these commands on the SnapVault primary storage system to restore data to a new qtree:

```
toaster> snapvault restore –s sv_weekly.40 –S oven:/vol/vol2/alpha
/vol/vol1/newq
```

SnapVault creates the new qtree, called newq, and transfers the backup data from the secondary storage system to this qtree.

When the transfer is complete, you can connect the data to the host by using a command such as:

```
snapdrive storage connect –fs /mnt/restore –hostvol SduDg/SduHv –lun
toaster:/vol/vol1/newq/lun1
```

All of the comments in the previous section about connecting to data are also applicable in this case. Because this method does not lock up any Snapshot copies, you can continue to use this data indefinitely. The data can coexist along with the live primary qtree (toaster:/vol/vol1/qtree1). It is also possible to make the new qtree as the source for further SnapVault updates.

4.3 Restoring data in place

If you no longer want the current data on the primary storage system, you may want to do in-place restore of the data. You may also need to do in-place restore if the volume does not have enough space to hold another copy of the data.

Before doing in-place restore, you must disconnect the existing storage from the host. This is because SnapVault does not support in-place restore of LUNs that are online.

```
snapdrive storage disconnect –fs /mnt/data
```

On the SnapVault primary storage system, first delete the existing qtree. After that, use this command to recover data in place:

```
toaster> snapvault restore –s sv_weekly.40 –S oven:/vol/vol2/alpha
/vol/vol1/qtree1
```

When the transfer is complete, you can connect the data to the host by using a command such as:

```
snapdrive storage connect –fs /mnt/data –hostvol SduDg/SduHv –lun
toaster:/vol/vol1/qtree1/lun1
```

After successfully restoring the data, you should use the `snapvault start -r` command to restart the SnapVault backup relationship between the primary and secondary qtrees (because you want to continue SnapVault protection of the data).

4.4 Restoring data from primary Snapshot copies

If the Snapshot copy of interest is available on the primary storage system, it is preferable to restore it from there. This works the same whether SnapVault is used or not. The primary storage

system allows all storage features, such as single-file SnapRestore®, LUN cloning, LUN resize, and so on.

Example command for restoring data from Snapshot copies from the primary:

```
snapdrive snap restore –fs /mnt/data –snapname 21march11pm
```

4.5 Using SnapVault for test and development

A variation of the configuration shown in Figure 1 is when two UNIX hosts are used, one at a primary site and another at a remote site. The remote site can use the same data that is created at the primary site for test and development purposes. This configuration can also be useful when it is necessary to present a read-only copy of the data at a remote location. Figure 2 shows this configuration.

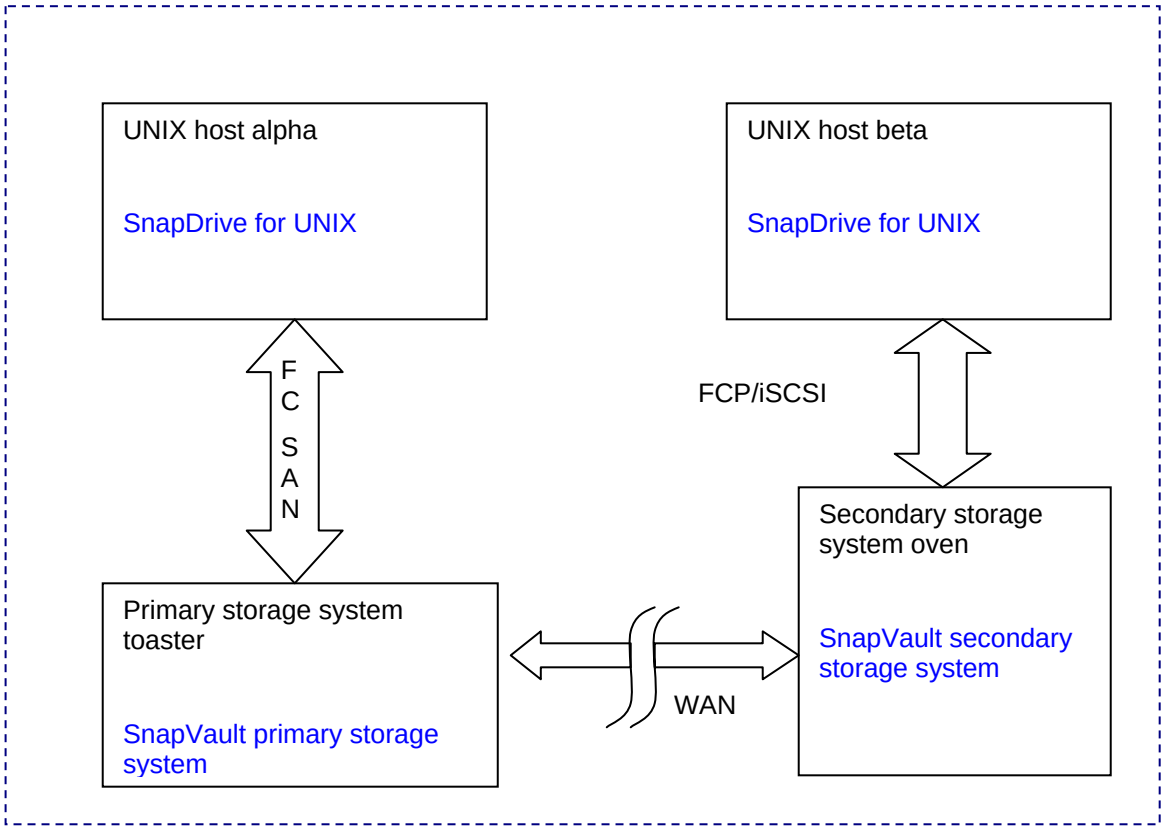

Figure 2) SnapDrive for UNIX with SnapVault configuration 2.

In this configuration, a second UNIX host (beta) is used at the remote site.

Most of the description from the previous section can be applied in this configuration. A few notable changes are:

- When checking the connectivity, check connectivity from the primary host to the primary storage system, and from the secondary host to the secondary storage system. Connectivity from the primary host to the secondary storage system is not necessary.
- No change is necessary in the mechanism for setting up the SnapVault relationship, provisioning, making Snapshot copies, and restoring Snapshot copies from the primary host.

- The SnapDrive for UNIX commands explained in section 4.1 should be executed on the secondary host (beta in this case) when accessing data from the secondary storage system.
- SnapVault enables you to delete older Snapshot copies from the primary storage system after they have been backed up to the secondary storage system (except for the last transferred Snapshot copy).

4.6 Using SnapVault with SnapDrive for UNIX in an NFS setup

SnapDrive for UNIX supports configurations in which the host is connected to the storage system on NFS. In such configurations, SnapVault can be used for backing up data or for giving a secondary host access to data for test and development purposes. Normal SnapVault mechanisms work fine in such configurations, because Snapshot copies that are initiated from the storage system are file-system consistent in an NFS environment. However, you may want to drive the Snapshot copy schedule from the host, if you need to perform host-specific actions before initiating a Snapshot copy.

The steps involved are similar to those in the previous section:

- Provisioning must happen on the storage system. Use commands like vol create to create a flexible volume, and enable NFS export permissions for the host (alpha). Then, from the UNIX host (alpha), issue a command like:
  `mount toaster:/vol/vol1/qtree1 /mnt/data`
- Example command for making Snapshot copies remains the same:
  `snapdrive snap create –fs /mnt/data –snapname 28sep649pm`
- To back up the Snapshot copy to the secondary system, use commands such as:
  `oven> snapvault update –s 28sep649pm /vol/vol2/alpha`

  Wait for the transfer to complete. Check the status of the transfer by using this command:

  `oven> snapvault status /vol/vol2/alpha`

  When the transfer is complete, make a Snapshot copy of the data by using a command such as:

  `oven> snap create vol2 28sep649pm`
- You can also use scripts invoked via cron to make scheduled Snapshot copies from the host.
- To create a clone volume based on the Snapshot copy, use this command on the storage system:
  `oven> vol clone create vol28sep –s volume –b vol2 28sep649pm`
- To access this cloned volume storage for read-write, use this command:
  `mount oven:/vol/vol28sep/alpha /mnt/remotedata`

Note: To allow root access, you may need to configure the export permissions on the storage system accordingly.

4.7 Host disk group spanning multiple storage systems

SnapDrive for UNIX supports disk groups that contain multiple LUNs, which can be spread over multiple storage systems. You can use such a configuration to obtain a high bandwidth to the storage subsystem.

It is possible to use SnapVault on such configurations, provided that the following conditions are met:

- The LUNs should be created inside qtrees.

- Each storage system qtree that is used on the primary storage must be a SnapVault primary storage system.
- You can use a common volume with multiple qtrees on the SnapVault secondary storage system that contains the backup data.
- Use the Consistency Groups feature of Data ONTAP and SnapDrive for UNIX to make a consistent Snapshot copy across the primary storage systems.
- Note the names of disk groups and logical volumes carefully—the same names should be used subsequently.
- For data recovery, create a FlexClone volume on the SnapVault secondary storage system based on the Snapshot copy of interest.
- Connect the corresponding LUNs to the UNIX computer (alpha or beta) by using `snapdrive storage connect –lun` commands.
- Note the device names of these LUNs (as they are mapped to the host), and then use these names to create disk groups.
- Mount the file system to the desired mount point on the host.
- The disk group configuration (number of LUNs, order of the LUNs, name of the disk group, name of the logical volume) must be identical to the original disk group.
- Some volume managers (for example, Veritas™ VxVM and Linux® LVM) do not allow an identical disk group to be connected to the system while the original disk group is still active. So this procedure can be used only when the original storage is lost or is on a different UNIX host.

Note: This section does not provide example commands. You can create your own commands based on the information provided in the previous sections. However, this can be a difficult task because of the complexities associated with using volume manager commands correctly.

5 Summary

SnapDrive for UNIX is a tool for storage provisioning and Snapshot management on UNIX systems connecting to NetApp FAS series storage systems. Data ONTAP has several unique features for data protection, including SnapMirror and SnapVault. This technical report details methods of using SnapDrive for UNIX and SnapVault together to back up and recover data.

6 Additional Resources

- Using SnapMirror with SnapDrive for UNIX: http://www.netapp.com/library/tr/3611.pdf
- SnapVault Best Practices Guide: http://www.netapp.com/library/tr/3487.pdf