Sybase ASE 12.5.x for UNIX®:

Integrating "quiesce database" Feature with Backup and Recovery Using a NetApp® Filer without Reboot

Giovanni Brignolo, Gangoor Sridhara, & David Yu | Network Appliance | March 2005 | TR-3166

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.

Network Appliance Inc.

Table of Contents

1. Purpose and Scope

This document covers the techniques for integrating the **quiesce database** feature provided in ASE for backing up and restoring a Sybase ASE 12.5.x server (henceforth called Sybase server) for a UNIX database when a Network Appliance™ filer is used for database storage. Specifically, this report covers the following topics:

- Using the quiesce database feature to take advantage of the filer Snapshot™ and SnapRestore™ functionality
- Backing up a database while the ASE is up and running (online backup)
- Backing up a database while the ASE is shut down (offline backup)
- Recovering a database when the transaction log is available
- Recovering a database when transaction log is not available

2. Key Advantages

The performance and reliability of the backup and recovery operations are critical to effective database operation. Network Appliance provides unique functionality that enhances performance and reliability in both backup and recovery procedures. Network Appliance's Snapshot technology is a key enabling feature for efficient and fast database backup and recovery operations. To take advantage of Snapshot technology, Sybase provided a new feature called **quiesce database** in the ASE 12.0 product release. With ASE, the **quiesce database** feature has been enhanced to support the log rollforward capability with Snapshots. With the **quiesce database** feature, consistent Snapshots can be taken by suspending the specified database(s) for update activities for a very short time. Updates will resume once the quiesced database(s) is/are released. In this way, while taking Snapshots, the database is available to the users. For more details about **quiesce database**, please refer to the Sybase ASE 12.5.x reference manual. For more information on how Snapshots work, see *File System Design for an NFS File Server Appliance* by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html). Those with NetApp on the Web (NOW) access can also check the following NetApp documentation, which covers Snapshots in detail. (NetApp on the Web is NetApp's online support site. This site is open to all current NetApp customers.)

Snapshots are critical because they allow the database administrator to quickly and easily create a read-only image of the entire file system, including the database data and log files. The Snapshot directory can be written to tape for archival while the database is being used. Actual backup-to-tape performance is thus of secondary importance, since the impact of this operation on the database's performance is negligible. With the **quiesce database** feature, there is no need to shut down the server to take Snapshots for backup operations. The archival operation does not impact the database availability and performance; this is normally an issue with non-filer storage systems and it is removed by using a filer. For backup operations, the database administrator (DBA) may choose either of the following methods:

- Suspend the database for update activities for the minimal amount of time while the server is running (online backup).
- Shut down and restart the server in a minimal amount of time (offline backup).

With Snapshots, an offline backup can be taken by shutting down the database, then taking a Snapshot. With Snapshots, the database will remain offline for only a short amount of time. Since the Snapshots take very little time, the server downtime will be minimized.

During online backup mode, a database is still available to users but is suspended for update activities. This results in the database being unavailable for write operations. Thus, getting the database out of suspended state is very important.

Since ASE 12.5.x provides a key feature, **quiesce database**, the DBA will find that the time required to take a backup using Snapshot is significantly reduced and there is no need to shut down the database server (for an offline backup) when using a Network Appliance filer.

Since the time required for the backup operation is significantly reduced, many DBAs may find that offline or online backups are more affordable when using a filer for database storage. The functionality of Snapshots provides the DBA with the ability to shut down the database server (for offline backup purposes) for only a few seconds, during which a Snapshot is taken, and then bring the database server up again.

Using online backup, you can also take Snapshots several times a day, and archive only one of them to tape. This offers the DBA additional flexibility. Effectively, you can take an online backup of your entire database every few hours. (*Note*: The amount of storage overhead associated with a Snapshot is based on the number of blocks that are different between the Snapshot and the active file system. As this increases, the size of the Snapshot's storage requirements will increase as well. Snapshots can be taken based on backup policy requirements. For more information, see 4.2. Backup Using the quiesce database Feature.

Sybase provided the initial implementation of the **quiesce database** command in the ASE 12.0 product release. This command allows the user to suspend the writes to the database(s) while an external copy is being performed; write suspension of the database is critical to ensure transactional consistency. However, in ASE 12.5.x, this feature has been enhanced with the capability to do log rollforward from Snapshots.

Snapshots provide key advantages for the restore operation. You can restore the data that you need (point-in-time data) from a Snapshot; in fact, you can restore the entire database, regardless of size, in a few minutes. This is accomplished with the SnapRestore function provided by the filer. If you save several days' worth of Snapshots, the chances are that you will never need to restore from tape at all, barring a catastrophic failure on the filer itself. Care should be taken to ensure all required Snapshots are maintained while restoring from the Snapshot using SnapRestore.

Consider the following scenario. A company is running a 200GB database on a UNIX platform, with storage on local disk. A glitch in one of the database server's CPUs results in corrupt data being written to the databases. As a result, the entire database must be restored from tape. Even at a rate of about 50GB per hour, it will take approximately four hours to restore the data.

Now consider the same scenario with storage on a filer. Since the DBA saved a Snapshot from a period prior to the failure, all that is required is to restore the data files from the Snapshot. Using the "snaprestore" command, this can be accomplished in only a few minutes, vastly improving mean time to recovery (MTTR) over that provided by the local disk solution. Starting with NetApp's Data ONTAP™ 6.0 release, rebooting after the restore is no longer required, which further improves data availability. However, if you are restoring the data of the root volume (`/vol/vol0`), a reboot of the filer is necessary.

The following is just a sample of times taken for the entire operation of quiesce database hold, Snapshot, and quiesce database release for update activities. These times do not include time taken for transaction load operation. The data is relevant to our test scenario.

Database Size	Data ONTAP Version	Time Taken for a Snapshot Including quiesce database	Time Taken for SnapRestore	Additional Remarks
500MB	6.1 - 7.x	Less than 10 seconds	Less than 10 seconds	
1GB	6.1 - 7.x	Less than 10 seconds	Less than 10 seconds	
20GB	6.1 - 7.x	Less than 10 seconds	Less than 10 seconds	

For more information on restoring a database using the "snaprestore" command, see 5.1. Restoring a Database with No Transaction Log Available and 5.2. Restoring a Database with Transaction Log Available.

3. Requirements and Assumptions

This technical report covers backup and recovery of a Sybase ASE 12.5.x database running under UNIX with database storage on a Network Appliance filer. We assume that you are familiar with Sybase ASE 12.5.x and the operation of NetApp filers. We also assume that you are familiar with the operation of your version of UNIX. All examples in this technical report are from Sybase ASE 12.5.x version running under Sun™ Solaris™ version 8.0 (or higher). The scripts contained in this paper may require significant modifications to run under your version of UNIX.

You must have your UNIX host configured to perform remote shell operations on the filer. This technical report assumes that the Sybase ASE 12.5.x server OS machine has this capability. For instructions on how to set up remote shell access to a filer, see http://now.netapp.com/knowledge/docs/ontap/rel53/html/sag/routine4.htm#1148067. (*Note*: This link requires NOW access. For more information, please see 2. Key Advantages.) The examples in this technical report use the following syntax:

```
rsh -l root filer2 SomeCommand
```

You can achieve the same results by issuing a similar command such as:

```
rsh -l sybase:sybase1 filer2 snap list
```

where `sybase:sybase1` is userid/password, `filer2` is the name of the filer, and `snap list` is the actual command.

As long as the `hosts.equiv` file on the filer is set up correctly, this is a secure operation. On our filer, the following entry is placed in the `hosts.equiv` file:

```
snaplink sybase
```

"snaplink" is the name of our UNIX machine where Sybase ASE 12.5.x server is running, and "sybase" is the name of our Sybase user account. This entry in the `hosts.equiv` file on the filer means that the Sybase user on snaplink is allowed to become a root on the filer for the purpose of running rsh commands. As long as this is properly administered, it does not create a security hole.

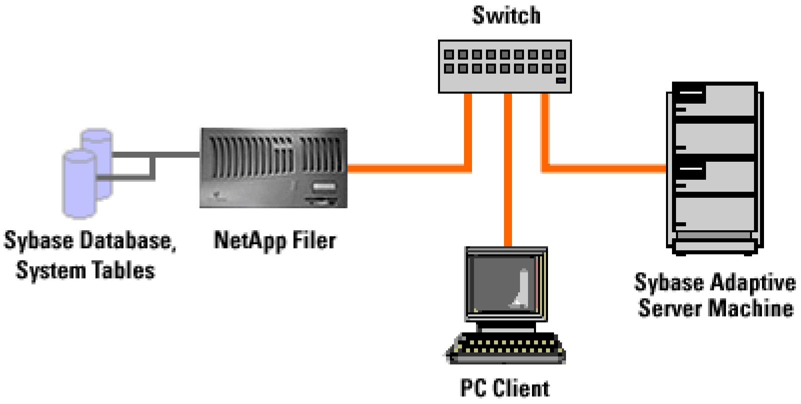
The sample scripts in this technical report assume the following:

- The name of the filer is `filer2`.
- The location of the database is `filer2:/vol/vol1/filer2`

- The name of the Sybase ASE 12.5.x server is "snaplink"
- The `filer2:/vol/vol1` directory is mounted on the Sybase ASE 12.5.x server machine at `/filer2`.

4. Backup

The following sections contain the techniques for backing up a Sybase ASE 12.5.x database stored on a Network Appliance filer.



4.1. Offline Backup

When a DBA decides to take an offline backup, it requires that the DBA shut down the database, back up all critical files using operating system commands, and then restart the database. However, this could be accomplished by using the **quiesce database** feature without the need to shut down the server.

Taking an offline backup using a NetApp filer is a simple and fast operation. It requires three scripts, which have the following functions:

Script Name	Function
`shutdown_server.sh`	Shuts down the Sybase ASE 12.5.x server.
`start_server.sh`	Brings up the Sybase ASE 12.5.x server.
`offlinebackup.sh`	A wrapper shell script that calls `shutdown.server.sh`, takes a Snapshot, calls `Start_server.sh`, and performs the backup-to-tape from the Snapshot. This script also does some housecleaning to keep four Snapshots online at a time. If the script is run every night as a cron job, then that four days' worth of Snapshots would be available at any given time. For more information, see 2. Key Advantages.

The `shutdown_server.sh`, `start_server.sh`, and `offlinebackup.sh` scripts are given as examples. Customize these scripts to your specific environment in order for them to work properly.

The following is the text of the `shutdown_server.sh` script:

```
 #!/bin/sh -f
# Now Shutdown the server
isql -Usa -P$PASSWORD -S$SYBASESERVERNAME << EOF
shutdown
go
EOF
```

You may want to wrap your shutdown command inside a shell script to shut down the server in normal mode. We recommend users not utilize the "`nowait`" option while using the "`shutdown`" command. The following is the text of the `start_server.sh` script. You can use this example to customize your script.

```
#!/bin/sh
#
# Adaptive Server name:      snaplink
# ASE page size (KB):        4k
# Master device path:        /filer2/release
# Error log path:  /filer2/release/ASE-12_5/install/snaplink.log
# Configuration file path:   /filer2/release/ASE-12_5/snaplink.cfg
# Directory for shared memory files:  /filer2/release/ASE-12_5
#
/filer2/release/ASE-12_5/bin/dataserver \
-ssnaplink \
-d/filer2/release/snaplink \
-e/filer2/release/ASE-12_5/install/snaplink.log \
-c/filer2/release/ASE-12_5/snaplink.cfg \
-M/filer2/release/ASE-12_5 \
```

The following is the text of the `offlinebackup.sh` shell script, and can be used as an example.

```
#!/bin/csh -f
# shutdown the Sybase ASE 12.5.x Server
shutdown_server.sh
#rename and delete old snapshots
#Delete Snapshot only when you no longer require that particular Snapshot
#This example is doing some housekeeping to manage Snapshots.
#In your own case, you may not wish delete your Snapshot
rsh -l root filer2 snap delete vol1 extdb3
#rename snapshot for retention
rsh -l root filer2 snap rename vol1 extdb2 extdb3
#rename snapshot for retention
rsh -l root filer2 snap rename vol1 extdb1 extdb2
#rename  current snapshot as previous snapshot for retention
rsh -l root filer2 snap rename vol1 newdb extdb1

# take a new snapshot and name it as newdb
rsh -l root filer2 create vol1 newdb

#the above steps can be customized
#start the Sybase ASE 12.5.x server again
# this is required as you are taking snapshot
#on offline mode. Server is down
start_server.sh
#perform your archival to tape whenever you need using ndmp
#or any other tools
# for example the following command in one line
# rsh -l root filer2 dump 0ufbln rst0a 63 /vol/vol1/release/.snapshot/newdb/
```

The options for dumping data from a filer to tape are amply documented in the Data ONTAP™ 7.0 Data Protection Tape Backup and Recovery Guide, which can be found online at now.netapp.com/NOW/knowledge/docs/ontap/rel70rc/pdfs/ontap/tapebkup.pdf (Note: This link requires NOW access. For more information, please see 2. Key Advantages.)

You will need to set the privileges in all of the above-mentioned shell scripts to allow them to be executed. The following commands will do this:

```
snaplink% chmod 750 offlinebackup.sh

snaplink% ls -l offlinebackup.sh
-rwxr-xr--   1 sybase  sybase       880 Sep 11 15:38 offlinebackup.sh
```

At that point, calling the `offlinebackup.sh` shell script will do the following:

```
$ ./offlinebackup.sh
     snaplink% shutdown_server.sh

     Server SHUTDOWN by request.
The SQL Server is terminating this process.
00:00000:00011:2005/03/29 06:32:26.38 server  SQL Server shutdown by request.
00:00000:00011:2005/03/29 06:32:26.38 kernel  ueshutdown: exiting
CT-LIBRARY error:
        ct_results(): network packet layer: internal net library error:
        Net-Library operation terminated due to disconnect
```

After this script runs without any errors, a new Snapshot will be created. This can be verified by using the `snap list` command.

On our system (a Sun Solaris server running Sybase ASE 12.5.x) the entire operation took less than a minute. *Note*: The size of the database will not appreciably affect this interval, since the time required to take a Snapshot is not dependent on the size of the file system, or the files within the file system. Rather, a Snapshot operation triggers a consistency point within the filer, and then simply copies the master inode of the file system to a new location. Thus, only about 4kB of data is actually copied. For more information, see *File System Design for an NFS File Server Appliance* by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html).

4.2. Backup Using the quiesce database Feature

The **quiesce database** feature allows the DBA (with filer root privileges) to take Snapshots while the server is running and available to users. The database is open for user access while doing backup. However, the quiesce database command suspends update activities for specified databases. For this reason, it is critical that the database remain in suspended mode for the absolute minimum period of time. Network Appliance affords the DBA the opportunity to reduce the interval during which the database is in suspended mode.

Sybase databases and log devices can be placed on the same volume on a filer. Using SnapRestore, the database can be restored up to the point of the last Snapshot. Sybase has provided the log rollforward capability with the Snapshot feature to support rolling the database forward from the restored Snapshot to the last committed transaction.

The steps for backing up and restoring the database involve the following procedure:

1. Login to the ASE using isql or other tools.

2. Issue the **quiesce database hold** command (needs system administrator [SA] role).

3. Take the Snapshot of the filer (needs admin privilege on filer).

4. Release the database hold using **quiesce database release** command.

5. Take transaction log dumps per your needs; name them in sequence.

6. Save these transaction log dumps on a different volume to preserve their presence through a restore.

7. If the DBA decides to have another Snapshot instead of log dumps, repeat steps 1 through 6.

Taking a backup while the server is running using a NetApp filer is a simple and fast operation. It requires the first four of the following scripts. These scripts have the following functions:

Script Name	**Function**
`Quiesce_use.sh`	This script calls `quiesce_hold.sh backup_snap.sh quiesce_rel.sh` and optionally `backup_dmp.sh`
`quiesce_hold.sh`	Suspends the specified databases for updates
`backup_snap.sh`	Takes backup and does housekeeping
`quiesce_rel.sh`	Releases the databases for update activities
`backup_dmp.sh`	Optional script to use dump command

To insert records we used the following script file (`tblcopy.sh`) that counts the total number of rows after inserting rows. This script was started before we executed the quiesce database command.

```
#!/bin/sh
$SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Ssnaplink << EOF
use userdb1
go
insert into titles select * from title
go
insert into title select * from titlea
go
select count(*) from titlea
go
select count(*) from title
go
select count(*) from titles
go
```

The following is the text of the `quiesce_use.sh` script:

```
     #!/bin/sh -f
     # This script suspends the database, take Snapshots, release database
     # and can also dump Snapshots to tape as a backup
     # calls quiesce_hold.sh
     quiesce_hold.sh
     backup_snap.sh
     quiesce_rel.sh
     # you can call backup_dmp.sh
     # backup_dmp.sh # this script can be run separately from another
           window # needs modification depending on your environment.
```

The following is the text of the `quiesce_hold.sh` script that you can customize to your environment:
The syntax of the `quiesce database` command is `quiesce database tag_name hold userdb1, userdb2 for external dump`

```
    #!/bin/sh -f
    # Quiesce the database and take a Snapshot
    # Make relevant changes to the following command syntax with correct database names
    # The syntax for quiesce database is given below
    # quiesce database tagname hold database1 [,database2] ... for external  dump
    $SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Ssnaplink << EOF
    quiesce database tagname hold userdb1 , master  for external dump
    go
     EOF
```

On our system, the `quiesce_hold` script generated this output resulting from suspending update activities:

```
00:00000:00013:2005/03/29 06:32:57.14 server
QUIESCE DATABASE command with tag tagname is being executed by process 13.
   00:00000:00012:2005/03/29 06:32:57.15 server
   Process 13 successfully executed QUIESCE DATABASE with HOLD option for tag tagname.
   Processes trying to issue IO operation on the quiesced database(s) will be
   suspended until user executes Quiesce Database command with RELEASE option.
```

The following is the text of the `backup_snap.sh` script:

```
  ##!/bin/sh -f
     #
     # Rename and delete old Snapshots
     # deletes the oldest Snapshot
     rsh -l root filer2 snap delete vol1 extdb3
     # rename the currently available oldest Snapshot for retention
     rsh -l root filer2 snap rename vol1 extdb2 extdb3
     # rename the next currently available older Snapshot for retention
          rsh -l root filer2 snap rename vol1 extdb1 extdb2
     # rename the most current Snapshot as older Snapshot for retention

     sh -l root filer2 snap rename vol1 newdb extdb1

     # Take a new Snapshot
     rsh -l root filer2 snap create vol1 newdb
     #rsh can also be executed  by specifying userid/passwd as given below
     #rsh -l sybase:sybase12 filer2 snap create vol1 newdb
     # where sybase:sybase12 is userid/password on the filer
```

In this example, one new Snapshot is created and three are retained. If this script is run as a cron job, you will have four days' worth of Snapshots at any time. It does some housecleaning to keep four Snapshots online. If the script is run every night as a cron job, then four days' worth of Snapshots would be available at any given time. Alternatively, you may want to take an online backup more frequently than once a day. Also you need to take frequent transaction log dumps for log rollforward capability. For more information, see 2. Key Advantages.

On our system the `backup_snap.sh` script generated the following output.

```
   creating snapshot...
```

The following is the text of `quiesce_rel.sh` script:

```
    #!/bin/sh
         $SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Ssnaplink << EOF
         quiesce database tagname release
         go
         EOF
```

On our system the `quiesce_rel.sh` script generated the following output:

```
00:00000:00013:2005/03/29 06:32:58.13 server
   Process 13 successfully executed QUIESCE DATABASE
   with RELEASE option for tag tagname.
```

The following is the text of the `backup_dmp.sh` optional script. This script can be used for archiving backups from Snapshots to tape.

```
 #!/bin/sh -f
# Perform your backup using dump
# or other operating system commands here.
# For example:
# rsh -l root filer2 dump 0ufbln rst0a 63 /vol/vol0/home/sybase/.Snapshot/newdb/
# make appropriate changes to these parameters to suit your environment
```

The process of suspending the database, taking Snapshots, and releasing the database procedure is complete.

The options for dumping data from a filer are amply documented in the Data ONTAP 7.0 Data Protection Tape Backup and Recovery Guide, which can be found online at: now.netapp.com/NOW/knowledge/docs/ontap/rel70rc/pdfs/ontap/tapebkup.pdf (Note: This link requires NOW access. For more information, please see 2. Key Advantages.)

You will need to set the privileges on all the above-mentioned shell scripts to allow them to be executed. The following commands will do this:

```
 snaplink% chmod 750 quiesce_use.sh quiesce_rel.sh backup_snap.sh
backup_dmp.sh quiesce_hold.sh
```

The output of 'ls -al' of quiese_use.sh was:

```
 snaplink% ls -al quiesce_use.sh
-rw-rw-r--   1 sybase       391 Mar 29 17:05 quiesce_use.sh
snaplink%
```

We have shown the output of these scripts, and the whole process took less than a minute.

To illustrate how the **quiesce database** feature works, we continued our database update activities after the database was released from suspended mode . Earlier we had issued a command to insert about 1 million rows to a table and another 75,025 records to another table in one of the user databases. While inserting the records, we issued a **quiesce database** command, took a Snapshot and released the quiesced database.

The `quiesce_hold.sh` script file and its output were provided earlier in this section.

Once all of the inserts were completed, the following output appeared on the screen:

```
       75027

(1 row affected)

 -----------
      1010747

(1 row affected)

 -----------
        75025

(1 row affected)
```

Now take transaction log dumps as needed. In our test case, we used a script file called `dumptran3demo3.sh` with the following contents (make sure that Sybase backup server is started before executing this script):

```
#!/bin/sh
$SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Ssnaplink << EOF
use master
go
dump tran userdb1 to "/home/sybase/userdb1log1"
go
```

The output of the above script was:

```
Backup Server session id is:  5.  Use this value when executing the
'sp_volchanged' system stored procedure after fulfilling any volume change
request from the Backup Server.
Backup Server: 6.28.1.1: Dumpfile name 'userdb10124105C74' section number 1
mounted on disk file '/home/sybase/userdb1log1'
Backup Server: 4.58.1.1: Database userdb1: 128520 kilobytes DUMPed.
Backup Server: 4.58.1.1: Database userdb1: 257042 kilobytes DUMPed.
Backup Server: 4.58.1.1: Database userdb1: 385564 kilobytes DUMPed.
Backup Server: 4.58.1.1: Database userdb1: 514086 kilobytes DUMPed.
Backup Server: 4.58.1.1: Database userdb1: 640864 kilobytes DUMPed.
Backup Server: 3.43.1.1: Dump phase number 3 completed.
Backup Server: 4.58.1.1: Database userdb1: 640874 kilobytes DUMPed.
Backup Server: 3.42.1.1: DUMP is complete (database userdb1).
```

On our system (a Sun machine running Sybase ASE version 12.5.x) the entire operation took less than a minute. *Note*: The size of the database will not appreciably affect this interval, since the time required to take a Snapshot is not dependent on the size of the file system, or the files within the file system. Rather, a Snapshot operation triggers a consistency point within the filer, and then simply copies the master inode of the file system to a new location. Thus, only about 4kB of data is actually copied. For more information, see *File System Design for an NFS File Server Appliance* by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html).

5. Restore

The examples concerning restore in this technical report assume that the data required is still stored in a Snapshot. If that is not the case (either because the Snapshot has been deleted or because the filer is not currently available), then you will need to perform 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 they are amply documented in the Data ONTAP™ 7.0 Data Protection Tape Backup and Recovery Guide which can be found online at: now.netapp.com/NOW/knowledge/docs/ontap/rel70rc/pdfs/ontap/tapebkup.pdf. (Note: This link requires NOW access. For more information, please 2. Key Advantages.)

5.1. Restoring a Database with No Transaction Log Available

In the event that transactions logs are not available, you may then restore from your most recent Snapshot (point-in-time recovery). Recovery is a manual process. The DBA should be intimately involved in each step of this process. For this reason, we do not supply a set of canned scripts to do a recovery. Instead, we illustrate the steps involved. The steps for point-in-time recovery are as follows:

1. Shut down the database with the `shutdown` command (if the database is still running).

2. Restore the databases (including the system databases) from the last Snapshot taken using the SnapRestore command or use a previous Snapshot for point-in-time recovery.

3. Restart the database.

To shut down the server, you can use the `shutdown_server.sh` script described earlier. To start the Sybase ASE 12.5.x server, you can use the script `start_server.sh`.

The `Recovery_complete.sh` script has the following lines:

```
# This is Recovery_complete.sh  to make sure "Recovery is Complete"
# Sleep a bit, or you'll detect wrong "Recovery complete"sleep 10
until
tail errorlog | grep -s "Recovery complete"
do
sleep 2
done
```

Next, you can execute the rsh command. On our system we executed a script called restoresnapshot.sh that had the following content:

```
#!/bin/sh

rsh -l root filer2 vol snaprestore vol1 -s newdb
#rsh -l sybase:sybase12 filer2 vol snaprestore vol1 -s newdb
./start_server.sh   &
 echo " Waiting for db recovery complete Message'
./Recovery_complete.sh
```

The output of the above command on our system was:

```
WARNING! This will revert the volume to a previous snapshot.
All modifications to the volume after the snapshot will be
irrevocably lost.

Volume vol1 will be taken offline briefly before coming back online.

Are you sure you want to do this? y

You have selected volume vol1, snapshot newdb

Proceed with revert? y
Volume vol1 is now offline.
exported /vol/vol1
Volume vol1 is now online
Volume vol1: revert successful.


.
.
.
00:00000:00001:2005/03/29 06:37:57.92 server  'bin_iso_1' (ID = 50)
00:00000:00001:2005/03/29 06:37:57.92 server  on top of default character set:
00:00000:00001:2005/03/29 06:37:57.92 server  'iso_1' (ID = 1).
00:00000:00001:2005/03/29 06:37:57.92 server  Master device size: 400 megabytes,
    or 204800 virtual pages. (A virtual page is 2048 bytes.)
00:00000:00010:2001/08/29 06:37:58.12 kernel  nconnect: t_rcvconnect,
    An event requires attention
```

The time required to do a recovery is minimal, only two or three minutes. This is true regardless of the size of the database, since the act of restoring a Snapshot is merely the copying of a single 4kB block. Beginning with Data ONTAP 6.0, a reboot of the filer is no longer required. This allows the DBA to quickly recover a database that can be very large in size (many gigabytes).

5.2. Restoring a Database with Transaction Log Available

The database will be restored using the most current or previous (point-in-time recovery) Snapshot. With the rollforward capability now integrated by Sybase with Snapshot, restoring a database with the availability of transaction log mode at the time of the failure presents a greater range of options. With the transaction log rollforward feature, the DBA can recover the database and roll it forward to the very last committed transaction using the transaction log. The full range of techniques covering all these options is far beyond the scope of this technical report.

Recovery is a manual process. The DBA should be intimately involved in each step of this process. For this reason, we do not supply a set of canned scripts to do a recovery. Instead, we illustrate the steps involved. You need to start the ASE server with the "-q" option to apply the transaction logs to the Snapshots restored.

The entire restore procedure is described as follows:

1. Bring down the ASE server if it's still running. If you are running other ASE servers on this volume on the filer, please shut down all the servers. If you restore while the ASE is up and running, the database will be marked as suspect and will need the DBA's attention. Also shut down the ASE servers using normal shutdown mode and do not use the nowait option . Bringing down the ASE server ungracefully may leave file with flocks not being released.

2. Telnet to the filer and issue the snaprestore command.

3. The filer restores to the point-in-time from the Snapshot. Restore the data files from the last Snapshot taken using the snaprestore command.

4. Start the server with "-q" option.

5. Apply the transaction logs by using the load transaction command. Do not bring the database online until all the required transaction logs have been applied. The sequence of the transaction log dumps is a must.

6. After all the logs have been applied, check the database and release it for user access.

This procedure completes the restore database procedure. Note that these transaction log dumps are relevant only with the previous Snapshot. If you take another Snapshot, you must maintain a new set of transaction log dumps to restore from the Snapshot taken most recently. As an example, if you take a Snapshot at 8 a.m. and name it sshot001_8am032905. Then let's assume that you take a transaction log dump at 9 a.m. (name it logdump001_9am032905), another transaction dump at 10 a.m. (name it logdump002_10am032905), yet another transaction log dump at 11 a.m. (name it logdump003_11am032905). Now let's assume that you decide to take another Snapshot at 12 noon (name it sshot002_12pm032905). At this time you have the option to recover the database to any one of the following points in time: 8 a.m., 9 a.m., 10 a.m., 11 a.m., and 12 noon. If you decide to recover up to 12 noon, simply restore from the Snapshot taken at 12 noon. There are no log dumps available for the Snapshot taken at noon. In case if you decide to recover only till 11 a.m., restore the Snapshot (sshot001_8am032905) and load transactions logdump001_9am032905, logdump002_10am032905, and logdump003_11am032905 in that particular order.

The script `shutdown_server.sh` can be used to shut down the server.

To restore the databases, you can execute a script called `restoresnapshot.sh`. In our test case, the contents of the `restoresnapshot.sh` script were:

```
 #!/bin/sh
 rsh -l sybase:sybase12 filer2 vol snaprestore vol1 -s newdb
./start_server.sh -q  &
echo " Waiting for db recovery complete Message'
sleep  20
 :
```

The above commands in our test case resulted in the following output:

```
WARNING! This will revert the volume to a previous snapshot.
All modifications to the volume after the snapshot will be
irrevocably lost.

Volume vol1 will be taken offline briefly before coming back online.

Are you sure you want to do this? y

You have selected volume vol1, snapshot sybdemo3

Proceed with revert? y
Volume vol1 is now offline.
exported /vol/vol1
Volume vol1 is now online
Volume vol1: revert successful.
Waiting for db recovery complete Message'

.
.
.
00:00000:00001:2005/03/29 06:37:57.82 server
   Recovering database 'userdb1'.
00:00000:00001:2005/03/29 06:37:57.86 server
   Database 'userdb1' is in QUIESCE DATABASE state.
   It will recovered as for LOAD DATABASE and left off line.
00:00000:00001:2005/03/29 06:37:57.90 server
   Checking external objects.
00:00000:00001:2005/03/29 06:37:57.91 server
   Database 'userdb1' appears to be in the process of being loaded;
   SQL Server will not bring it online automatically. Use the ONLINE DATABASE
   command to bring this database online.
00:00000:00001:2005/03/29 06:37:57.92 server  Recovery complete.
00:00000:00001:2005/03/29 06:37:57.92 server
   SQL Server's default unicode sort order is 'binary'.
00:00000:00001:2005/03/29 06:37:57.92 server  SQL Server's default sort order is:
00:00000:00001:2005/03/29 06:37:57.92 server  'bin_iso_1' (ID = 50)
00:00000:00001:2005/03/29 06:37:57.92 server  on top of default character set:
00:00000:00001:2005/03/29 06:37:57.92 server  'iso_1' (ID = 1).
00:00000:00001:2005/03/29 06:37:57.92 server
   Master device size: 400 megabytes, or 204800 virtual pages.
   (A virtual page is 2048 bytes.)
00:00000:00010:2005/03/29 06:37:58.12 kernel nconnect:
   t_rcvconnect, An event requires attention
```

To show that the quiesced database is not available for user access, we issued the following command:

```
1> use userdb1
2> go
Msg 950, Level 14, State 1:
Server 'userdb1', Line 1:
Database 'userdb1' is currently offline. Please wait and try your command again
later.
```

Now start loading the transaction logs to the database(s). On our test case, we used a script called `tlog5.sh` after starting the Sybase backup server. The `tlog5.sh` file looked like:

```
#!/bin/sh
$SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Ssnaplink << EOF
use master
go
load tran userdb1 from "/home/sybase/userdb1log1"
go
online database userdb1
go
EOF
```

We verified the data and updated the table, and verified that transactional consistency was maintained by running `rowcount.sh` script. The output from this script was:

```
 75027

(1 row affected)

     1010747

(1 row affected)

       75025

 )
```

6. Conclusions

A Network Appliance filer offers the Sybase ASE 12.5 DBA compelling advantages in terms of backup and recovery. Use of Snapshots, combined with conventional backup-to-tape techniques, can dramatically optimize the Sybase ASE 12.5.x database backup operation. Retaining a number of online Snapshots allows the DBA to restore the database without the need to restore from tape in many circumstances. Backup and recovery performance is dramatically improved over conventional directly attached disk (JBOD or RAID) configurations.

7. Caveats

Sybase supports the use of network-attached storage only in the context of Network Appliance filers. For more information regarding use of Network Appliance filers with Sybase, see http://www.netapp.com/tech_library. You can also find support on Sybase's Web site for NetApp filers. Please recheck this page as we are planning to add several documents regarding Sybase products on Network Appliance filers. However, NetApp has tested this configuration on some versions of UNIX other than Sun Solaris, such as HP/UX, Red Hat Linux, and IBM AIX, and has not tested with all the combinations of hardware and software options available on these platforms. There may be significant differences in your configuration that will alter the procedures necessary to accomplish the objectives outlined in this paper. You may need to make changes to these scripts to ensure that they work in your environment. If you find that any of these procedures do not work in your environment, please contact NetApp technical support.

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