Applications for Writeable LUNs and LUN Cloning in Oracle® Environments

Toby Creek | June 2003 | TR 3266

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 will explore the use of writable Snapshot image-backed LUNs and LUN cloning in conjunction with an Oracle database. These features, available in Network Appliance™ storage systems running Data ONTAP™ 6.4 or later, enable space-efficient and expedient creation of cloned Oracle databases for reporting, test environments, and database reorganization.

2) Assumptions

For the information contained in this report to be useful to the reader, several assumptions are made:

The reader has at least intermediate Windows® 2000 administration skills and has access to the administrator login or an account that is a member of the local or domain administrator's group, or:

The reader has at least intermediate Solaris™ or other UNIX® system administration skills and has access to the root user account.

The reader has at least intermediate Oracle9i™ administration skills and has the Oracle9i installation media or has Oracle9i already installed on the target system.

The reader has at least intermediate knowledge of Network Appliance filer administration, and has administrative access to the filer via the command-line interface as well as FilerView® software.

The filer and host have the required licenses necessary to perform the activities outlined in this document.

The target system must have the required protocol interconnects to perform the activities outlined in this document.

For the purposes of this report, filer administration is performed via telnet for clarity. FilerView or scripts utilizing rsh can also be used where applicable. Microsoft® Windows implementations should use the SnapDrive console or command line interface.

3) Additional Resources

For additional information that will assist the reader in utilizing the procedures detailed here, see the following Network Appliance technical reports:

For Microsoft Windows platforms:

Oracle9i™ for Windows® 2000: Integrating with a NetApp® Filer in a SAN Environment

Oracle9i™ for Windows® 2000: Backup and Recovery Using a NetApp® Filer in a SAN Environment

For Sun™ Solaris:

Oracle9i™ for UNIX®: Integrating with a NetApp® Filer in a SAN Environment

Oracle9i™ for UNIX®: Backup and Recovery Using a NetApp® Filer in a SAN Environment

4) Infrastructure

The specifics of the hardware configuration are not of central importance to this paper, so hardware specifics will not be discussed here. Each scenario as presented will have specific infrastructure

requirements, as will the protocol and family of filer used. Software platform differences will affect the specific commands used in each procedure. Infrastructure requirements will be addressed in each section.

5) Cloning Databases Using Writable Snapshot Copies

Database cloning is often accomplished by making a complete copy of all database files to an alternate location. In cases where the clone is to be run on the same machine, the database is renamed by recreating the control files.

Writable Snapshot software-backed LUNs afford the database administrator the ability to clone a database by taking a Snapshot, creating a Snapshot image-backed LUN, and connecting that LUN to a host. Because of the structure of the WAFL® file system, the clone actually shares unchanged blocks with the original database; no data copy is involved. Changed blocks are written to new areas on disk, and the WAFL inode pointers are updated with this location information.

Clone databases created in writable Snapshots should be designed with temporary use in mind. Reporting databases, which normally have a short lifespan, are an excellent application for writable Snapshot copies.

The procedure for creating clone databases using writable Snapshot copies will depend on whether the Snapshot image-backed LUN will be connected to a second host. If the Snapshot is connected to the same host, the database name must be changed before the database can be started. If a second host is used, the database name can remain the same.

5.1) Cloning Databases Using Writable Snapshot Copies

Because Oracle cannot allow two databases with identical names to run on the same host, and changing the database's name using data files in hot backup mode is not possible without recovering the database first, only cold backups can be used to create single-host clones.

The procedure for creating a clone from a cold backup is detailed in the following steps:

1. Back up the control file to the user trace directory using the `alter database backup controlfile to trace` command in SQLPlus.
2. Shut down the target database cleanly. Issuing `shutdown abort` is not an acceptable shutdown method.
3. Create the Snapshot of all database LUNs using a distinctive name. At a minimum, the Snapshot must contain the redo logs and all database files. If the database runs in archivelog mode, the archived logs will be needed.

```
filer> snap create /vol/dbvol snap1
```

4. Create a new parameter file for the new Oracle instance. It is best to copy the original database parameter file and update the parameters to reflect the new drive letters or pathnames to be used.

5. Copy the control file trace from the user dump directory and edit the file to reflect the new database name. Be sure to insert the `set` command in front of the database name, and verify that all data file locations point to the clone database drive letter or path name.
6. Create Snapshot copy-backed LUNs using the `lun create –b` command from the filer command line and map the LUN to the host, as shown in the example. Optionally, the "-o noreserve" switch can be used to disable space reservation to conserve disk space at the expense of database robustness.

```
      filer> lun create –b /vol/dbvol/.snapshot/snap1/dblun.lun
-o noreserve /vol/dbvol/writeable.lun
      filer> lun map /vol/dbvol/writeable.lun dbhost
      lun map: auto-assigned dbhost=1
```

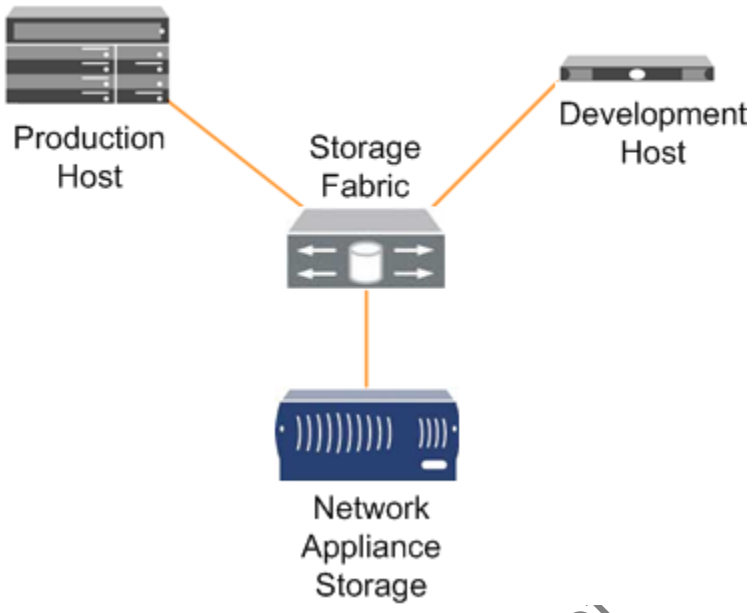
7. On the host, mount or connect the writable LUNs to the appropriate drive letter or path name. Ensure that the path used matches the contents of the new parameter file and control file creation script.
8. Run SQLPlus, start the database in NOMOUNT mode, and recreate the control files using the modified trace file from the prior step.
9. The database can now be opened.

5.2) Second-Host Clone Creation

Creating a clone database using a second host is a more robust method for creating reporting databases. Since the clone database runs on a second host, the instance does not have to compete for resources with the production instance on the same machine. Also, the second host could be on an entirely different front-end network from the production machine, allowing more flexibility in determining access control. In addition, the database name need not be changed; a different net service name can be configured on the clients, but the Oracle System Identifier (SID) need not be changed.

The two hosts must be connected to the same storage fabric as the filer. The particular protocol used in the fabric is of no consequence; either iSCSI or Fibre Channel fabrics can be employed.

A basic diagram of the architecture is shown in the diagram:



In this section, it is assumed that the original database is run in archivelog mode, as all production database instances should be for recoverability. The production system can remain online during the cloning process; Oracle's hot backup mode will be used to make the clone. Scripting can be employed for all of these steps to automatically refresh the reporting database.

The creation of a clone database for a second host is detailed in the following steps:

1. Back up the control file to the user trace directory using the `alter database backup controlfile to trace` command in SQLPlus.
2. Using the `alter tablespace … begin backup` command in SQLPlus, and put all of the tablespaces in hot backup mode. For more details on creating a script to perform this function, see Oracle 9i: Backup and Recovery in a SAN Environment.
3. Create the Snapshot of all database LUNs using a distinctive name. At a minimum, the Snapshot must contain the redo logs and all database files. Additionally, any archive logs created during the backup will be needed.

```
filer> snap create /vol/dbvol snap1
```

4. Using the `alter tablespace … end backup` command in SQLPlus, take all of the tablespaces out of hot backup mode.
5. Copy the Oracle parameter file from the production database. Some parameters, such as the database buffer cache, shared pool size, and large pool size can be adjusted for the target system if necessary.

6. Create Snapshots backed by LUNs using the `lun create –b` command from the filer command line. Optionally, the "-o noreserve" switch can be used to disable space reservation to conserve disk space at the expense of database robustness.

```
filer> lun create –b /vol/dbvol/.snapshot/snap1/dblun.lun
–o noreserve /vol/dbvol/writeable.lun
filer> lun map /vol/dbvol/writeable.lun dbhost2
lun map: auto-assigned dbhost2=1
```

7. On the host, mount or connect the writable LUNs to the appropriate drive letter or pathname. Ensure that the path used matches the contents of the new parameter file and control file creation script. If this is not possible, the control files can be recreated using the control file trace created previously.
8. Start the database in MOUNT mode.
9. Issue `alter database end backup` to take all tablespaces out of hot backup mode.
10. Issue the `recover automatic database` from the SQLPlus command line. At this point, Oracle may prompt for the archived logs needed to complete the recovery.
11. Issue `alter database open` to make the database available.

At this point, redo log archiving can be disabled if necessary to save disk resources. Any additional setup, such as granting user access, can also be performed before you make the database available for use.

Considerations for Volume Management Software

When using volume management software to manage LUNs, additional procedures and considerations may need to be addressed. Though volume management software is outside of the scope of this report, this section will enumerate some of these issues.

Most volume management software, including the Veritas Volume Manager, writes signatures to the LUN that contain unique host identification information, as well as information about other disks in the group. Additional steps will be required to "import" these groups onto the same host (where possible) or the second host. Importing a LUN volume backed by a Snapshot onto the same machine as the original should not be attempted; Veritas does not support the procedure required.

When volume management software is used to stripe or mirror LUNs, all LUNs in a volume set must be created as Snapshot copy-backed LUNs.

6) Seeding Test Environments Using LUN Cloning

Creating test environments for developer access can be a resource-consuming process. These environments are usually not created with great frequency, and can have a long lifespan.

Any Snapshot for which there exist Snapshot-backed LUNs will be marked as busy. They cannot be destroyed until that LUN is destroyed. Additionally, the Snapshot software-backed LUN is created in the active file system, which means that it will appear in subsequent Snapshot copies. It is possible to create a dependency on a Snapshot such that it cannot be destroyed without destroying all subsequent Snapshots. LUN cloning solves this potential problem by making the LUN permanent in the active WAFL® file system, removing the dependency on the Snapshot. Once cloned, the Snapshot slot is freed for reuse in the backup rotation.

LUN cloning is performed using the `lun clone` commands from the filer command line. LUNs are cloned to the same pathname as the original. LUN clones inherit their space reservation settings from the Snapshot copy-backed LUN from which they are cloned.

The process for creating cloned databases using LUN cloning is the same as previously outlined for reporting databases, with one additional step: the LUN backed by a Snapshot is cloned after the last step, as in the following command example.

```
filer> lun clone start /vol/dbvol/writeable.lun
filer> lun clone status /vol/dbvol/writeable.lun
lun clone: Done 330112 of 1311298 blocks (25% complete).
Wed Mar 26 10:55:00 EST [vdisk_admin:info]: Cloning completed on lun
/vol/dbvol/writeable.lun
```


After the `lun clone` command completes, the LUN is permanent and is no longer dependent on the Snapshot.

7) Reorganizing Read-Mostly Databases Using Writable Snapshots and LUN Cloning

Read-mostly environments generally are data warehouses, where data is loaded in batches and queries are run to extract data. In this case, read-mostly environments could also be considered to be databases where data is inserted only during certain periods of the day.

The LUN manipulation features of Data ONTAP can be employed to allow online reorganization of these types of environments, incurring minimal downtime since the database can be available while the reorganization takes place. The database is then briefly taken offline to complete the switchover to the reorganized copy.

The steps required to implement this type of reorganization are detailed in the following steps:

1. Clone the LUNs containing the database data as shown in the previous sections.
2. If possible, alter all tablespaces in the original database read-only, to avoid data changes that would be lost.
3. Reorganize the database. Reorganization can include dropping or resizing data files, migrating data between tablespaces, or any other data manipulation needed.
4. Once reorganization is complete, record the LUN paths and their associated numbers using the `lun show –v` command.

5. Shut down the original database, then unmap and destroy its associated LUNs.
6. Map the cloned LUNs to the original LUNs, preserving the LUN numbering. If reorganization involved combining or removing LUNs, ensure that the host configuration is updated to reflect the change.

8) Sample Scripts

For reference, sample scripts are included in this report to aid in the creation of automated solutions for database cloning. These scripts may require extensive modification before being used. The scripts will require remote administrative access to both the filer and the source database. In the case of Microsoft Windows, the second host will also need to have administrative privileges on the host where the source database resides. This is normally established through domain credentials.

clonelun.ksh:

```
#!/bin/ksh
# put oracle in hot backup mode
su - oracle -c "sqlplus /nolog @begin_backup.sql"

# take the snapshot
rsh filer snap create /vol/dbvol snap1

# take the database out of hot backup
su - oracle -c "sqlplus /nolog @end_backup.sql"

# create the snapshot-backed LUN
rsh filer lun create -b /vol/dbvol/.snapshot/snap1/dblun.lun -o
noreserve /vol/dbvol/writeable.lun
rsh filer lun map /vol/dbvol/writeable.lun dbhost2

# mount the filesystem on the local machine
fsck -y /dev/dsk/c2t1d0s1
mount /dev/dsk/c2t1d0s1 /u01

# startup Oracle
su – oracle –c “sqlplus /nolog @startup.sql”
```

clonelun.bat:

```
rem put oracle in hot backup mode
sqlplus /nolog @begin_backup.sql

rem take the snapshot
sdcli snap create -m prodhost -s snap1 -D O

rem take oracle out of hot backup
sqlplus /nolog @end_backup.sql

rem create the snapshot backed lun and connect to current machine
sdcli snap mount -m devhost -r prodhost -s snap1 -k O -d O

rem start the Oracle instance
sqlplus /nolog @startup.sql
```

```
begin_backup.sql:

      connect sys/sys@source as sysdba;
      alter tablespace system begin backup;
      alter tablespace sys_undots begin backup;
      alter tablespace users begin backup;


end_backup.sql:

      connect sys/sys@source as sysdba;
      alter tablespace system end backup;
      alter tablespace sys_undots end backup;
      alter tablespace users end backup;


startup.sql:

      connect / as sysdba;
      startup mount;
      alter database end backup;
      recover automatic database;
      alter database open;
```


9) Conclusions

NetApp Snapshot technology has always given the database administrator a robust and extremely fast backup mechanism. Storage virtualization provided by Data ONTAP now gives the system administrator and database administrator powerful tools to quickly and efficiently create cloned databases for a variety of uses.

10) Caveats

For more information, see the resources cited at the beginning of this report. You can also find support on Oracle's Web site for NetApp storage. However, NetApp has not tested this procedure with all of the combinations of hardware and software options available. There may be significant differences in your configuration that will alter the procedures necessary to accomplish the objectives outlined in this paper. If you find that any of these procedures do not work in your environment, please contact the author immediately.