TECHNICAL REPORT

Nimble Storage – Oracle Backup and Recovery Guide

Document Revision

Table 1.

Date	Revision	Description
1/23/2013	1.0	Initial Draft

THIS TECHNICAL REPORT IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCUURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Table of Contents

Introduction

The purpose of this technical white paper is to describe the different techniques of backup and recovery for Oracle database using Nimble Storage.

Tuning Oracle database performance is beyond the scope of this paper. Please visit www.oracle.com for *Oracle Performance Tuning Guide* for more information in tuning your database.

Audience

This guide is intended for Oracle database solution architects, storage engineers, system administrators and IT managers who analyze, design and maintain a robust database environment on Nimble Storage. It is assumed that the reader has a working knowledge of network design for iSCSI SAN, and basic Nimble Storage operations. Knowledge of Oracle Linux operating system, Oracle Clusterware, and Oracle database is also required.

Scope

Architecting an Oracle environment for backup and recovery for a business SLA is not an easy task. There are many factors that need to be considered during this phase. These factors include, but not limited to database server operating system, file system or Oracle ASM, volume management such as LVM or VxVM, Oracle RAC or non-RAC, backup and recovery methods such as RMAN or User-managed, and disk storage. Every storage vendor has their own snapshot feature but not all are created equal. Some claim that their snapshot feature is space efficient, but has poor performance. Others claim good performance, but poor in space efficiency. Nimble Storage snapshot and clone features are space efficient, are instantaneous without impacting performance. This paper discusses how to implement Oracle database backup using Nimble Storage snapshot and zero-copy clone in a production environment.

Other technical reports on Oracle databases cover complementary topics such as best practices, storage setup for Oracle and test and development environments using zero-copy clones.

Nimble Storage Snapshot and Zero-Copy Clone Features

Instantaneous Point-in-Time Snapshots

Take point-in-time copies, which do not require data to be copied on future changes (redirect-on-write) -- all the data in snapshot remains as is. This does not impact performance and facilitates fast restores without copying data. Customers benefit from a single, simple storage solution for primary and secondary data, frequent and instant backups, fast restores and significant capacity savings.

Zero-Copy Clones

Instantly create full functioning copies or clones of volumes without copying any data. Customers get great space efficient and performance on cloned volumes, making them ideal for database recovery, test, development, and staging Oracle databases.

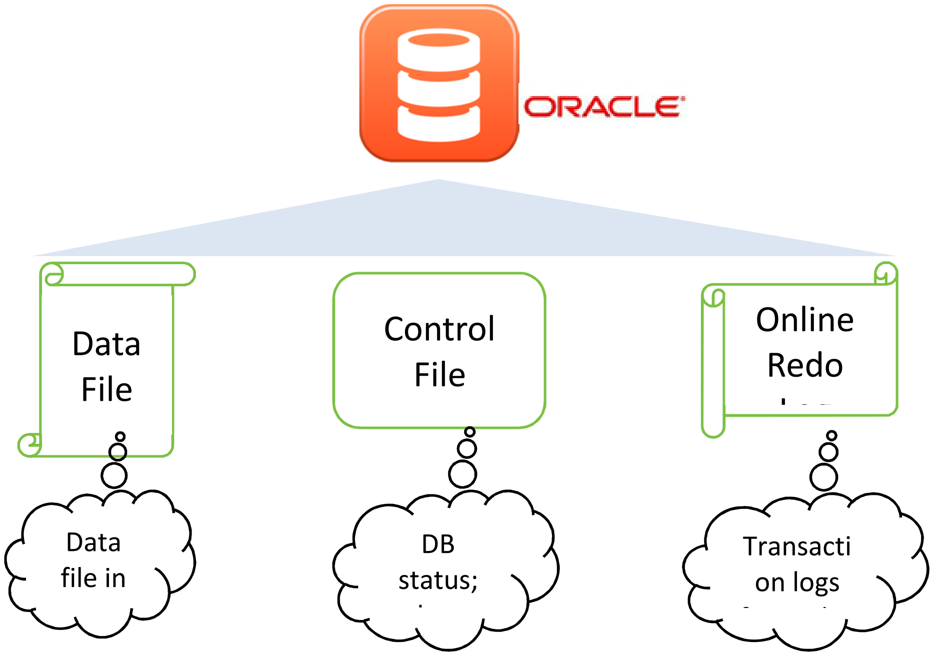
Oracle Database Overview

Oracle Physical Storage Structures

- Data File
    - Contains all database data in units called data blocks.
- Control File
    - This contains the record of the physical structures of the database and its status. Several types of information stored in the control file are related to backup and recovery:
        - Database information (RESETLOGS SCN and time stamp)
        - Tablespace and data file records (filenames, data file checkpoints, read/write status, offline ranges)
        - Information about redo threads (current online redo log)
        - Log records (log sequence numbers, SCN range in each log)
        - A record of past RMAN backups
        - Information about corrupt data file blocks
    - The recovery process for data files is in part guided by status information in the control file, such as the database checkpoints, current online redo log file, and the data file header checkpoints for the data files. Loss of the control file makes recovery from a data loss much more difficult.
- Online Redo Logs
    - Redo logs record all changes made to a database's data files. Each time data is changed in the database, that change is recorded in the online redo log first, before it is applied to the data files in accordance to ACID-compliance. Because the redo log contains a record of all changes to the data

files, if a backup copy of a data file from some point in time and a complete set of redo logs from that time forward are available, the database can apply changes recorded in the redo logs, in order to re-construct the data file contents at any point between the backup time and the end of the last redo log. However, this is only possible if the redo log has been preserved. Therefore, preserving the redo logs is a major part of most backup strategies.

The first level of preserving the redo log is through a process called archiving. The database can copy online redo log groups that are not currently in use to one or more archive locations on disk, where they are collectively called the archived redo logs. Without archived redo logs, the database backup and recovery options are severely limited. The database must be taken offline before it can be backed up, and if you must restore your database from backup, the database contents are only available as of the time of the backup (RPO). Restructuring the state of the database at a point in time between backup is impossible without the archived logs.



Storage Setup for Oracle Databases

Oracle Backups Overview

Backups in Oracle can be achieved via Recovery Manager (RMAN) or User-managed backup. RMAN is the preferred backup solution if you are using Oracle ASM and also require logical recovery (e.g. tablespace or datafile). Both methods are supported by Oracle and are fully documented at http://www.oracle.com/technetwork/indexes/documentation/index.html.

Recovery Manager (RMAN)

○ RMAN, a tool (with command-line client and Enterprise Manager interfaces) that integrates with sessions running on the Oracle server to perform a range of backup and recovery activities, as well as maintaining a repository of historical data about your backups. RMAN only back up database files needed for efficient recovery in the event of a failure. The following file types are supported by RMAN.

- Data files, and image copies of data files
- Control files, and image copies of control files
- Archived redo logs
- The current server parameter file
- Backup pieces, containing other backups created by RMAN

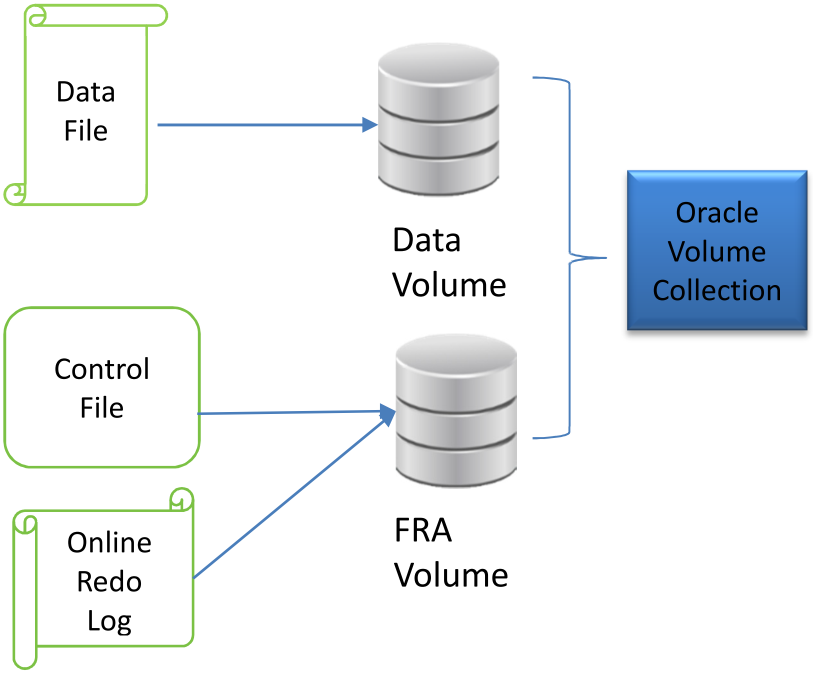
Note: Although the database depends on other types of files for operation, such as network configuration files, password files, and the contents of Oracle home directory, these files cannot be backed up with RMAN. You must use some non-RMAN backup solution for any files not in the above list. This is where the Nimble Storage snapshot is useful.

- User-Managed Backup
  - The traditional user-managed backup, where you directly manage the files that make up your database with a mixture of SAN snapshot, host operating system commands, and SQL*Plus backup and recovery-related capabilities.

Using Nimble Storage Snapshot and Zero-Copy Clone for Oracle Backup

Backing up Oracle databases with Nimble snapshot is simple. Both Oracle RMAN and User-Managed backup methods will work with Nimble snapshot capability. Both methods are discussed below.

First step with either method is the process of creating a Nimble Storage Volume Collection. The volume collection can be created using the Nimble Storage management interface and should comprise of the ‘Data’ and ‘FRA’ volumes from the production server.



Create a Volume Collection for Oracle Backup

Zero-Copy Clone with RMAN

RMAN always stores its RMAN repository of metadata in the control file of each target database on which it performs operations. Optionally, you can use RMAN with a recovery catalog. In this case, RMAN maintains an additional repository of metadata in a set of tables in a separate recovery catalog database. The two methods are:

- RMAN with a recovery catalog database
- RMAN with control files

In either case running RMAN directly on the production server is costly and can impact performance during the database backup window. During the RMAN backup process, it can consume a lot of server resources such as CPU and memory. You need to have a dedicated backup window (usually at night) to avoid resource contention with normal operations. This can create more work for an on-call DBA if the backup did not complete successfully. With the Nimble Storage snapshot and clone features, the DBA can back up the database during the day and not worrying about resource contention by offloading it to a backup server.

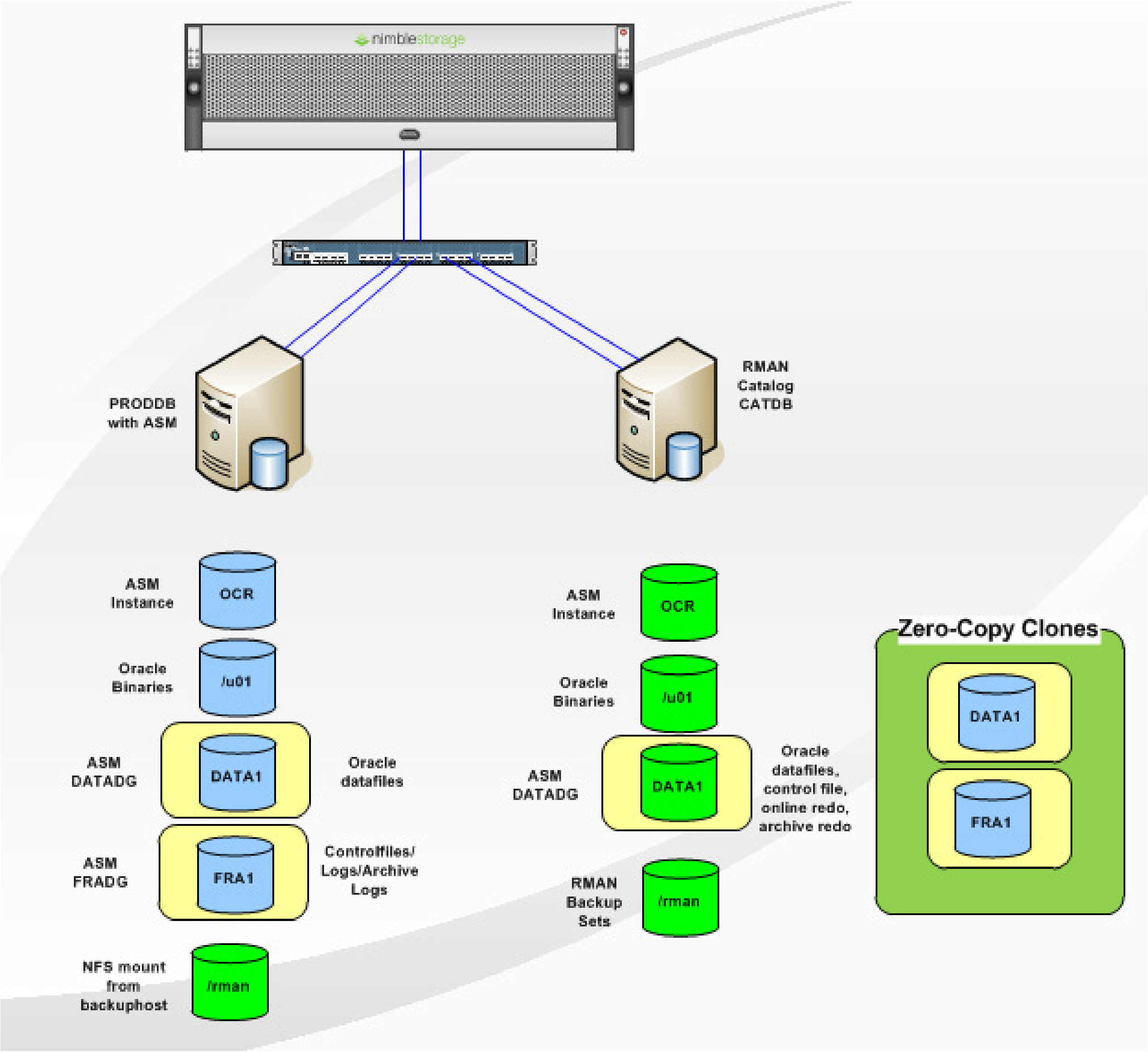
RMAN with Recovery Catalog Database

A recovery catalog database is a schema used by RMAN to store metadata about one or more Oracle databases. Typically, you store the catalog in a dedicated database. A recovery catalog provides the following benefits:

- A recovery catalog creates redundancy for the RMAN repository stored in the control file of each target database. If the target control file and all backups are lost, then the RMAN metadata still exists in the recovery catalog.
- A recovery catalog centralizes metadata for all your target databases and store metadata history much longer than the control file.

Assumption:

The procedure below assumes that you have 2 hosts. HOST1 is the production Oracle database and HOST2 is the RMAN catalog database. You would like to offload the RMAN backup process to disk from production by using the Nimble zero-copy clone feature. This will allow you to run backups at any time without contending for server resources. Here is an example:



Step-by-Step

Throughout this section, two hosts are illustrated: The production host which runs the production Oracle database. The backup host, which can be a physical or virtual machine will be used to perform the RMAN backups of the production database to disk (/rman filesystem).

1. On the production host, put the production database in backup mode.

Sample script to put the production database in backup mode.

```
su – oracle <<EOF
export ORACLE_SID=”SID”
sqlplus “/as sysdba” <<END
whenever sqlerror exit1
alter system archive log current;
alter database begin backup;
END
EOF
```

2. Take a snapshot of a Volume Collection. In this case, the Volume Collection contains the two volumes (DATA1 and FRA1) of the production server. You can use the Nimble management interface or the CLI to take the snapshot. Below is an example of running the CLI on the backup host with the Volume Collection named “PRODDB” and the snapshot named “PRODDB_BACKUP”.

```
[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} volcoll - -snap PRODDB - -snapcoll_name
PRODDB_BACKUP
```

3. On the production host, take the production database out of backup mode.

Sample script to take the production database out of backup mode.

```
su – oracle <<EOF
export ORACLE_SID=”SID”
sqlplus “/as sysdba” <<END
whenever sqlerror exit1
alter database end backup;
alter system archive log current;
END
EOF
```

4. Create a zero-copy clone from the snapshot and present the clone volumes to the backup host. You can use the Nimble Storage CLI to accomplish this task. Run this command for both volumes (DATA1 and FRA1).

```
[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -clone DATA1 - -snapname PRODDB_BACKUP - -
clonename DATA1-clone - -apply_acl_to volume - -initiatorgrp “backuphost”

[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -clone FRA1 - -snapname PRODDB_BACKUP - -
clonename FRA1-clone - -apply_acl_to volume - -initiatorgrp “backuphost”
```

5. On the backup host, run the iSCSI discovery and login commands.

```
[root@backuphost ~] # iscsiadm –m discovery –t st –p <iSCSI discovery IP>

[root@backuphost ~] # iscsiadm –m node –T <target volume iqn> - -login
```

6. On the backup host, scan for ASM disks.

```
[root@backuphost ~] # /etc/init.d/oracleasm scandisks
```

7. On the backup host, list the ASM disks for verification.

```
[root@backuphost ~] # /etc/init.d/oracleasm listdisks
DATA1
FRA1
```

8. On the backup host, mount the ASM disk groups as grid user.

```
[grid@backuphost ~] $ srvctl start diskgroup –g DATADG
[grid@backuphost ~] $ srvctl start diskgroup –g FRADG
```

9. On the backup host, start mount the production database to run RMAN backup as oracle user.

```
su – oracle
[oracle@backuphost ~] $ export ORACLE_SID=PRODDB
[oracle@backuphost ~] $ sqlplus / as sysdba
SQL> startup mount;
SQL> alter database end backup;
SQL> exit
```

10. On the backup host, run RMAN backup to disk as oracle user. In this example, the backup server has a volume created on the Nimble array and is mounted under /rman directory.

First, use your favorite editor and create a RMAN backup .rcv file (proddb.rcv) as follow.

```
{
configure device type disk parallelism 4;
allocate channel c1 type disk maxopenfiles 10 format ‘/rman/%U.bak’;
allocate channel c2 type disk maxopenfiles 10 format ‘/rman/%U.bak’;
allocate channel c3 type disk maxopenfiles 10 format ‘/rman/%U.bak’;
allocate channel c4 type disk maxopenfiles 10 format ‘/rman/%U.bak’;
backup as compressed backupset database plus archivelog;
}
```

Next, execute the RMAN backup

```
[oracle@backuphost ~] $ rman target / CMDFILE proddb.rcv LOG /tmp/rman_backup.log
```

11. On the backup host, catalog the RMAN backup pieces with the catalog database as Oracle user.

First, use your favorite editor to create a RMAN script file (proddb.sql) as follow. Do this for all backup pieces that were generated by the RMAN backup. In this example, there were 6 backup pieces were created by RMAN.

```
run
{
catalog backuppiece “/rman/01nt3ebt_1_1.bak”;
catalog backuppiece “/rman/02nt3ebt_1_1.bak”;
catalog backuppiece “/rman/03nt3ebt_1_1.bak”;
catalog backuppiece “/rman/04nt3ebt_1_1.bak”;
catalog backuppiece “/rman/05nt3ebt_1_1.bak”;
catalog backuppiece “/rman/06nt3ebt_1_1.bak”;
}
```

Next, execute the RMAN catalog script by logging into the catalog database with the target database as proddb.

```
[oracle@backuphost ~] $ rman catalog=rman/rman_password@catalogdb target=sys/sys_password@proddb
```

```
RMAN> @/proddb.sql
RMAN> resync catalog;
RMAN> exit
```

12. Once the catalog completes, we need to clean up the clone volumes on the backup host. Shutdown the PRODDB database on the backup host first.

```
su – oracle
[oracle@backuphost ~] $ export ORACLE_SID=PRODDB
[oracle@backuphost ~] $ sqlplus / as sysdba
SQL> shutdown abort;
SQL> exit
```

13. Unmount the ASM disk groups as grid user.

```
[grid@backuphost ~] $ srvctl stop diskgroup –g DATADG
[grid@backuphost ~] $ srvctl stop diskgroup –g FRADG
```

14. Delete ASM disks from the backup host after unmounting the disk groups

```
[root@backuphost ~] # /etc/init.d/oracleasm deletedisk DATA1
[root@backuphost ~] # /etc/init.d/oracleasm deletedisk FRA1
```

15. Remove the clone volumes from the backup host. You can use the Nimble CLI to accomplish this task.

```
[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -removeacl DATA1-clone - -apply_acl_to volume - -
initiatorgrp “backuphost”

[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -removeacl FRA1-clone - -apply_acl_to volume - -
initiatorgrp “backuphost”
```

16. Delete the clone volumes from the Nimble array. You can use the Nimble GUI or the CLI to accomplish this task. Here is the CLI command.

```
[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -offline DATA1-clone - -force

[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -offline FRA1-clone - -force

[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -delete DATA1-clone

[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} vol - -delete FRA1-clone
```

17. On the backup host, log out of iSCSI for the two deleted volumes. Repeat the command if you have multiple iSCSI networks.

```
[root@backuphost ~] # iscsiadm –m node –T <iqn> -p <iSCSI IP> - -logout
```

18. Delete the snapshot that was taken for the RMAN backup from step #2 above.

```
[root@backuphost ~] # ssh admin@${NIMBLE_ARRAY} snapcoll - -delete PRODDB_BACKUP
```

19. On the backup host, clean up the iSCSI devices for the two volumes.

```
[root@backuphost ~] # cd /var/lib/iscsi/nodes

[root@backuphost ~] # rm –r <iqn>
```

RMAN with Local Control Files

Offloading RMAN backups using the Nimble Storage snapshot and zero-copy clone feature with the local control files is similar to the procedure above EXCEPT for step #11. Instead of cataloging the RMAN backup pieces with the catalog database, you want to catalog the backup pieces with the control files on the production host. Below is the instruction for step #11.

Note: The filesystem containing the backup sets (/rman) must be NFS export to the production server.

Revised Step 11 for RMAN backup with local control files.

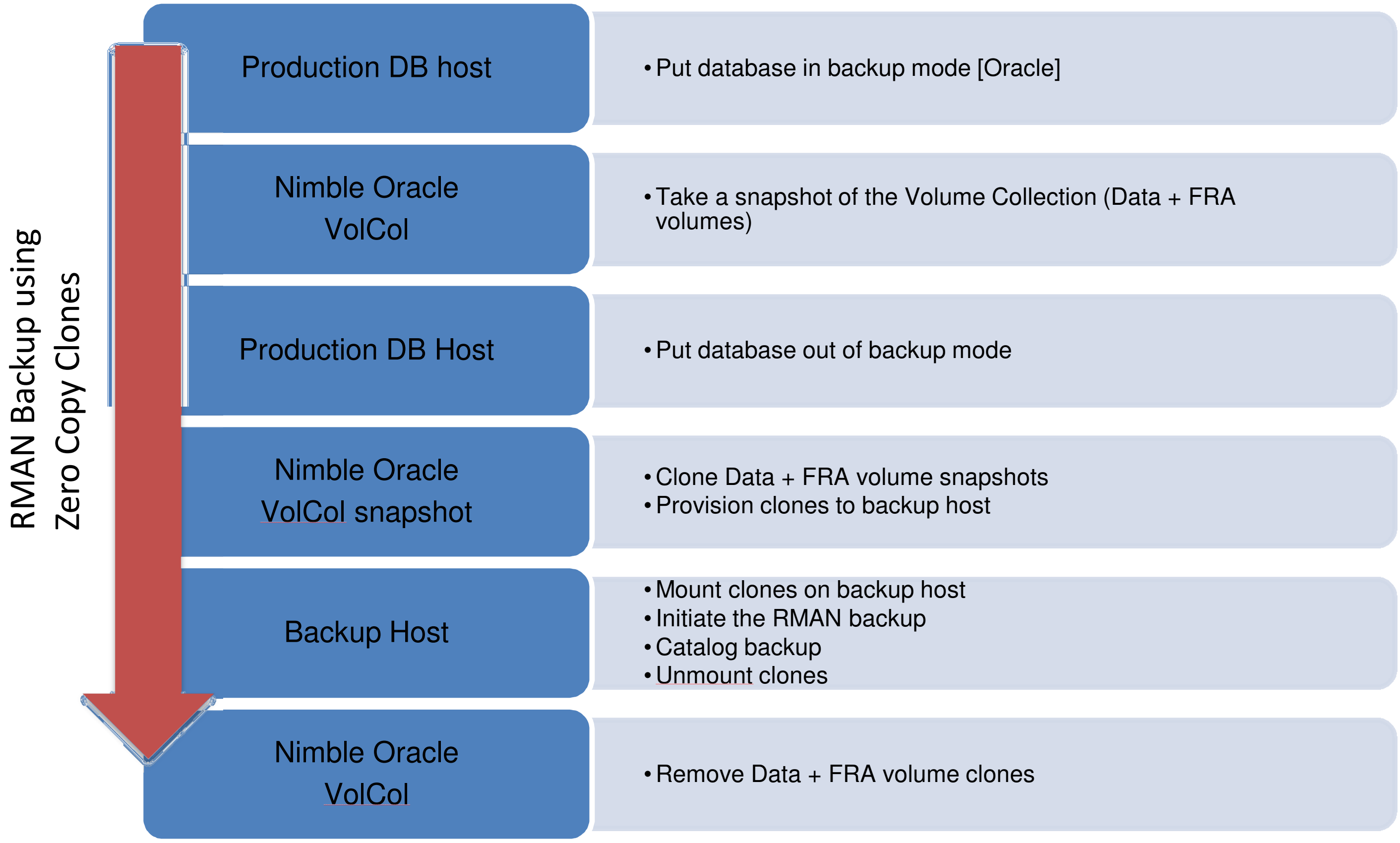
First, on the production host, use your favorite editor to create a RMAN script file (proddb.sql) as follow. Do this for all backup pieces that were generated by the RMAN backup. In this example, there were 6 backup pieces were created by RMAN.

```
run
{
catalog backuppiece “/rman/01nt3ebt_1_1.bak”;
catalog backuppiece “/rman/02nt3ebt_1_1.bak”;
catalog backuppiece “/rman/03nt3ebt_1_1.bak”;
catalog backuppiece “/rman/04nt3ebt_1_1.bak”;
catalog backuppiece “/rman/05nt3ebt_1_1.bak”;
catalog backuppiece “/rman/06nt3ebt_1_1.bak”;
}
```

Next, execute this RMAN catalog script on the production server.

```
[oracle@productionhost ~] $ rman target /
RMAN> @/proddb.sql
RMAN> resync catalog;
RMAN> exit
```

Here is a flow chart summarizing the process of RMAN backup using zero-copy clones


RMAN Backup using Zero Copy Clones
Production DB host
Put database in backup mode [Oracle]
Nimble Oracle VolCol
Take a snapshot of the Volume Collection (Data + FRA volumes)
Production DB Host
Put database out of backup mode
Nimble Oracle VolCol snapshot
Clone Data + FRA volume snapshots
Provision clones to backup host
Backup Host
Mount clones on backup host
Initiate the RMAN backup
Catalog backup
Unmount clones
Nimble Oracle VolCol
Remove Data + FRA volume clones

Zero-Copy Clone with User-Managed Backup

The traditional user-managed backups usually comprise of operating system commands such as cpio, cp, LVM mirror/split, etc. This approach only works if you run your production database on a file system such as EXT, JFS2, VxFS, etc. Nimble Storage's snapshot capability makes LVM mirror/split and commands obsolete.

Below is the discussion on how to use the Nimble Storage snapshot and zero-copy clone with Oracle user-managed backup. If you have a database running on EXT file system and would like to back up that database, you can simply follow the instructions above to take a snapshot of all volumes that comprise your database. Here are some questions you need to understand when using snapshot with file system.

1. For recovery, are you planning on mounting the snapshot volume back to the same host while the original volume is still mounted? For example, you'd like to recover a file from the snapshot volume.
2. If yes, is your existing volume being used by any type of volume management such as LVM or VxVM?

If you're not using volume manager such as LVM or VxVM, then mounting the snapshot back to the same host with the original volumes intact is possible. If you use LVM or VxVM and want to mount the snapshot volume with the original volume intact, then you need to make sure they are the latest version (i.e. LVM v2, VxVM 5+) which will allow you to do just that. You also need to have an in-depth knowledge of how LVM or VxVM behave in order to accomplish this task.

For example, if you have a single volume on your Oracle database server without the use of LVM or VxVM, you can use the Nimble Storage snapshot and zero-copy clone feature on that volume if you have a need to do any type of recovery such as recover a file. All you have to do is create a zero-copy clone from a snapshot and present it to the Oracle database server. You can use the same procedure above to take a snapshot and create a clone. Once the clone volume is presented to the Oracle server, run the iSCSI discovery and login commands. Next, create a directory so you can mount this volume and recover any missing file. When you're done, you can clean up the clone volume using the above procedure.

Oracle Database Recovery

Implementing a backup strategy solely depends on how you would do a recovery. There are a few methods to recover a database:

1. Full database recovery
2. Partial database recovery (logical recovery)
3. Point-in-time recovery (i.e. System Change Number (SCN) or time)

Using the Nimble Storage snapshot and zero-copy cloning features for database recovery is the most effective and time-saving approach. Below is the discussion for the database recovery with Oracle ASM and file system.

Recover Oracle Database Running on ASM

If your database is running on ASM, there are two ways to do a database recovery.

1. Restore and recover the database with the snapshots
2. Restore and recover the database with Oracle RMAN

Restore and Recover the Database with Snapshots

Using the native Nimble Storage snapshot feature works best if you want to restore and recover the whole database in minutes. For example: Let say that a database is on a single ASM diskgroup called “DATADG”. In this DATADG diskgroup, there are a total of 4 ASM disks. Each ASM disk is a volume on the Nimble array. If I want to recover the whole database using one of the available snapshots, at a high level, I would simply do the following:

1. Shutdown the current database

```
[root@mktg02 asm]# srvctl stop database -d <db_name>
```

2. Stop the ASM DATADG diskgroup

```
[root@mktg02 asm]# srvctl stop diskgroup -g <diskgroup_name>
```

3. Take the 4 original volumes offline on the Nimble Storage. Use the Nimble management interface or CLI (shown below

```
[root@mktg02 asm]# ssh admin@${NIMBLE_ARRAY} vol - -offline <volume_name> - -force
```

4. Restore the 4 original volumes with the selected snapshot. In this case, I use the snapshot called “testsnap”.

Run the same commands for the 4 volumes.

```
[root@mktg02 asm]# ssh admin@${NIMBLE_ARRAY} vol - -restore <volume_name>  - -snapname testsnap
```

5. Take the 4 original volume online.

Run the same commands for the 4 volumes.

```
[root@mktg02 asm]# ssh admin@${NIMBLE_ARRAY} vol - -online <volume_name>
```

6. Verify ASM disks

```
[root@mktg02 asm]# /etc/init.d/oracleasm scandisks
```

```
[root@mktg02 asm]# /etc/init.d/oracleasm listdisks
```

7. Start ASM disk group

```
[root@mktg02 asm]# srvctl start diskgroup -g <diskgroup_name>
```

8. Start database

```
[root@mktg02 asm]# srvctl start database -d <db_name>
```

Restore and Recover the Database with Oracle RMAN

Using Oracle RMAN for database recovery works best for logical recovery such as a single tablespace or a datafile. Recover the whole database using Oracle RMAN can take a long time depending on how big the database is. If the database is backed up via RMAN and and a single tablespace needs to be recovered, use the following process:

1. Make sure the production server can access the RMAN backup sets. In this case, the RMAN backup process is offloaded on the backup server as described in the previous section. The RMAN volume must be mounted using NFS on the production server.
2. As Oracle user, log into RMAN. Specify the proper RMAN login string depending on how the backup was taken – that is whether it was using a catalog database or a control file.
3. Once in RMAN, recover the database object via the restore and recovery command.

Conclusion

Data protection of your critical Oracle database made simple and efficient using Nimble Storage snapshots and zero-copy clones. By implementing backup and recovery the right way, RTO and RPOs can be significantly improved keeping applications and users up and running.

Nimble Storage, Inc.

211 River Oaks Parkway, San Jose, CA 95134

Tel: 877-364-6253) | www.nimblestorage.com | info@nimblestorage.com