White Paper

Creating an Architecture for Cloning Oracle Databases on Demand

Fred Gomez, NetApp
May 2010 | WP-7097

EXECUTIVE SUMMARY

Creating and maintaining multiple Oracle® database copies for development and test, business intelligence, and other purposes can be a storage- and time-intensive process. This paper describes an alternative approach using NetApp FlexClone® technology to create a “clone-on-demand” environment in which multiple thin clones of a production database can be created in minutes using a fraction of the storage space of full copies. These clones can improve your development and business intelligence processes while also simplifying DR testing. Cloning processes can be fully integrated with data masking using SnapManager® for Oracle software.

TABLE OF CONTENTS

1 INTRODUCTION

In almost any Oracle environment there is a need for multiple copies of production databases. These copies are used to support a variety of activities, such as:

- Application development and test (dev/test)
- Quality assurance
- Business intelligence
- Problem reproduction and root cause analysis

The traditional approach to copying a production database is a heavyweight activity. Since every copy requires 100% of the storage capacity of the original database, keeping many copies places a major burden on storage resources. Getting necessary approvals, locating storage capacity, and copying the data also typically take a long time to accomplish. Performance of the production database can also be impaired while the copy is taking place. While it may be possible to perform testing or QA with only a subset of your production data, this can lead to inconsistent results, and other activities will require the complete data set.

Because full database copies are an expensive resource, they are always in short supply and often not as current as you would like them to be. DBAs, developers, testers, and line-of-business specialists all queue up waiting for limited resources. Such bottlenecks inevitably increase time to deployment/market, slow business processes, and decrease staff productivity.

This white paper describes an alternative approach using NetApp FlexClone copies. NetApp FlexClone allows you to architect a “clone-on-demand” environment in which you can create a clone of your production database at any point in time, whenever it is needed, without tying up large amounts of storage, wasting time, or impeding production.

2 NETAPP TECHNOLOGY OVERVIEW

This section provides a brief overview of the NetApp technology involved in this solution. If you are familiar with NetApp aggregates, flexible volumes (FlexVol® volumes), Snapshot™ copies, and NetApp FlexClone technology you can skip this section. If you’d like to learn more about these technologies as they pertain to database applications, refer to NetApp Technical Report 3373: Data ONTAP 7G—The Ideal Platform for Database Applications.

NetApp Snapshot technology has been built into the NetApp operating environment since the first release. The NetApp Write Anywhere File Layout (WAFL®) file system—which organizes all NAS and SAN data written to disk—never rewrites a block to the same location. As a result, creating a Snapshot copy of an entire volume or part of that volume is fast, simple, and efficient.

To make a Snapshot copy, a copy of the root block pointers is created and saved. Whenever a block in the root is changed, that block is written to a new location. The original block, in its original location, is saved as part of the Snapshot copy as illustrated in Figure 1. As a result, Snapshot copies only consume disk space as changes are made to the volume, and maintaining multiple Snapshot copies has no performance impact. Another advantage of this approach is that you can quickly revert to a saved Snapshot copy in the event of a database error using NetApp SnapRestore®, and then replay your redo logs to restore the database from that point in a matter of minutes.

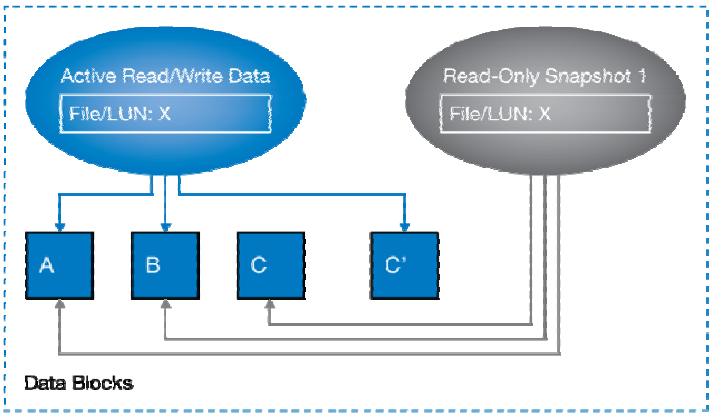

Figure 1) NetApp Snapshot technology.

NetApp Data ONTAP® 7G and later operating environments introduce the additional concepts of aggregates, flexible volumes (FlexVol volumes), and flexible copies (FlexClone copies).

FlexVol volumes and FlexClone copies are contained inside an aggregate. A single aggregate can be as large as 16TB and is composed of a large number of physical disks grouped into RAID groups using the NetApp dual-parity RAID implementation, RAID-DP®. RAID-DP is an advanced, cost-effective failure-/error-protection solution that protects against double disk failure within a single RAID group.

Using FlexVol, you can grow and shrink volumes at will as long as the aggregate has enough space. Because aggregates contain a large number of disk spindles, performance bottlenecks are greatly reduced; even the smallest FlexVol volume within an aggregate is spread across all spindles.

Using FlexClone, you can create a space-efficient, writable clone of any FlexVol volume or LUN. As of Data ONTAP 7.3.1, you can also clone individual files inside a FlexVol volume (NAS environment) or LUN (SAN environment). FlexClone essentially creates the read/write equivalent of a Snapshot copy, so additional space is only consumed as changes are made to either the original volume or the clone. Because no data blocks are copied, a FlexClone volume can be created in a matter of seconds regardless of the size of the volume. Traditional full copies, by comparison, can take hours to complete.

NetApp developed its SnapManager for Oracle solution to help simplify the use of these and other features in Oracle environments. SnapManager for Oracle integrates NetApp capabilities with Oracle so that Oracle data is in a consistent state (hot backup mode) when you are backing up, cloning, or replicating it.

3 LAYING OUT YOUR ORACLE DATABASE ON NETAPP STORAGE

The recommended layout for an Oracle database in a NetApp environment is somewhat different than a typical Oracle layout that you might use in conjunction with RMAN. This layout is designed to achieve a number of things:

- Facilitate Snapshot copies for quick backup.
- Limit downtime should a hardware failure occur.
- Simplify replication and disaster recovery.
- Enable cloning on demand.

NetApp storage systems used to support databases are usually configured in a clustered, active-active configuration. One storage controller has active control over half the storage, while the other storage controller controls the other half. In the event of a failure, either controller can take control of the other's storage.

The primary goal in laying out an Oracle database across the two storage controllers is to minimize the possibility that any type of failure, no matter how significant, can prevent the database from being restarted. The preferred layout is illustrated in Figure 2.

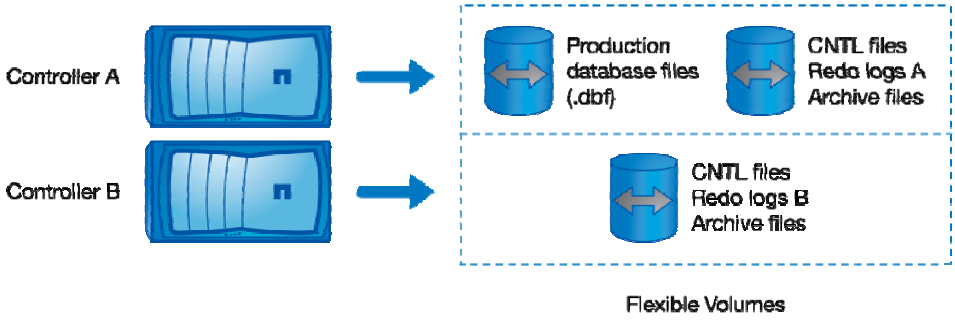

Figure 2) Simple layout for an Oracle database on clustered NetApp storage.

As you can see, the objective is to put the database files (.dbf files) in a FlexVol volume on one storage controller and have two copies of the CNTL file, redo logs, and archive files in two separate FlexVol volumes, one on each storage controller. This way, even if one side goes down entirely, you can recover using the resources on the other side. Another reason for separating the database files from the redo logs is that it gives you the flexibility to create separate Snapshot and replication schedules for each.

If you have a larger database, you might choose to distribute the database files in two or more FlexVol volumes spread across both storage controllers for optimum performance. (The specifics of how you choose to do this may depend on the type of data you are storing.) Similarly, if you have multiple databases you will want to balance them across the two controllers using the same basic scheme illustrated in Figure 2 for each database. Another optimization that is used in some environments is to create a separate FlexVol volume for temp space to avoid creating Snapshot copies of temporary and volatile database files that don't need to be backed up.

In this configuration, Snapshot copies become the first line of defense against database failure. You can establish a schedule of regular and frequent Snapshot copies that will minimize the amount of redo logs that must be replayed in the event that a restore becomes necessary.

You can use SnapManager for Oracle to set up these schedules. When a Snapshot copy is scheduled, the SnapManager software automatically puts the database in hot backup mode to provide consistency, take the Snapshot copy, and then return the database to normal operation in a matter of seconds. Once a Snapshot copy has been created, it can be backed up to tape, vaulted to secondary storage using NetApp SnapVault® software, or replicated to secondary storage using NetApp SnapMirror® software for disaster recovery.

4 REPLICATION OPTIONS FOR CLONING ON DEMAND

Although it is possible to use FlexClone to make clones of your active production database on primary storage, most people choose to make clones from a replicated copy of the database to avoid any possible interference with the performance of the production database that might result from I/O traffic to its clones.

If a NetApp storage system is the secondary storage target, NetApp SnapMirror software provides an efficient and convenient mechanism for replicating the database on a continuing basis. Unlike many other replication solutions, NetApp SnapMirror does not require the target storage system to be the same as the source. You can choose a standalone configuration with economical SATA disk drives if that satisfies your I/O requirements.

The mirrored copy can be in a remote location to provide disaster recovery in addition to serving as the basis for creating database clones on demand.

NetApp SnapMirror provides two options:

- **Volume SnapMirror** is used to replicate entire FlexVol volumes and will also replicate any Snapshot copies made of those FlexVol volumes.
- **Qtree SnapMirror** can be used to replicate individual qtrees. (A qtree is a structure you create inside a FlexVol volume. Its primary purpose is to control quotas within that subcontainer.) Volume Snapshot copies are not replicated by qtree SnapMirror.

If you use the database layout described in the previous section—with separate FlexVol volumes for different database components—you probably won't have to be concerned with replicating subsets of a volume. However, in this case the important differentiator is whether or not Snapshot copies are replicated.

With volume SnapMirror, if you create a clone of a Snapshot copy on your secondary storage system and that Snapshot copy is subsequently deleted from the primary system, the next replication update will fail. This is because the Snapshot copy on the secondary system cannot be deleted when one or more clones of it exist and you want the clone(s) to continue to function. The mirror has to be identical to the source, and, in the case of volume SnapMirror, this includes source Snapshot copies as well.

In practice it turns out to be much more convenient to simply create a qtree within each FlexVol volume to serve as a container for database files and use qtree SnapMirror to replicate that qtree. Once this is done, you can have different Snapshot schedules on primary and secondary storage.

On primary storage you might have a schedule in which Snapshot copies of all associated database FlexVol volumes are made on an hourly basis and then one Snapshot copy of each for each day of the week is retained.

On secondary storage you might find it more convenient to create a Snapshot schedule tailored to your cloning needs by retaining Snapshot copies that represent historical versions of the database that are weeks, months, or even years old.

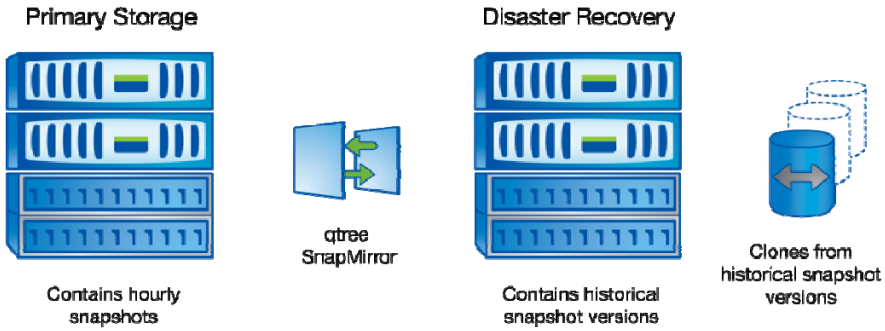

Figure 3) Replication using qtree SnapMirror. Different Snapshot schedules can be maintained on primary and secondary storage.

5 CREATING THE CLONE YOU WANT AT THE RIGHT POINT IN TIME

Once you've created your replication schedule and an appropriate Snapshot schedule on both primary and secondary storage, it becomes a simple process to create a database clone at any point in time for any purpose.

You can make a Snapshot copy of the active mirror of the database and clone it to create a database copy that is completely up to date. You can get an historical version of your database by cloning the appropriate Snapshot copy from the library of saved Snapshot copies you created.

You can also clone the nearest Snapshot copy in time and then restore it back to a specific point in time by replaying redo logs. This can be advantageous for Oracle Real Application Testing (which requires synchronization between the test database and transactions captured from production) or for problem reproduction and root cause analysis. You simply clone the most recent Snapshot copy made before the event in question (a crash, etc.) and then use Oracle tools to replay the log files up to the exact point when the problem occurred to analyze the event in real time.

Once again you can use NetApp SnapManager for Oracle to control and manage your database cloning.

6 SUPPORT FOR DR TESTING

Another significant advantage of this approach to cloning on demand is that it facilitates DR testing. With most replication solutions, you can't use the replicated images to test your DR processes without disrupting the replication schedule. You might need to do a back-up before you start to test and, you may have to start the replication over from scratch.

With NetApp cloning on demand, you can create clones of all of your replicated database volumes and then test your DR processes using those clones. Ongoing replication proceeds without disruption and, when you are finished, you simply delete the clones to release any additional storage space that was consumed.

7 DATA MASKING

Any time you use data from your production database for testing or similar purposes, you have to think about the need to protect sensitive information such as names, addresses, account information, credit card information, and so on. Regulations around the world, including the Sarbanes Oxley Act of 2002 in the United States and the European Union's Data Protection Initiative, now mandate the protection of confidential or sensitive information contained in corporate databases.

Many companies already have third-party data masking tools or homegrown scripts to accomplish the masking process. In addition, the Oracle Data Masking Pack provides regulatory compliance through consistent and rule-based application of masking formats with a rich and extensible format library that supports a variety of masking options.

The difficulty that arises is making sure that these tools have been appropriately applied to every database copy. Fortunately, it's straightforward to incorporate data masking into your cloning-on-demand environment. You accomplish this by creating one or more "master masked clones" as illustrated in Figure 4.

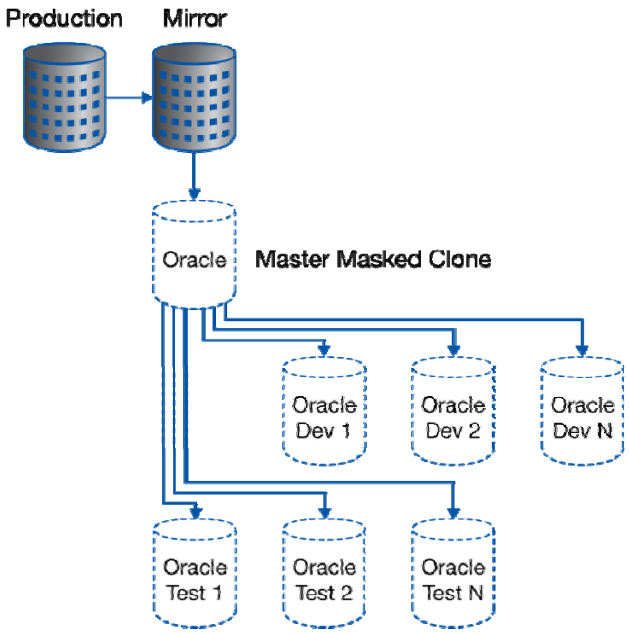

Figure 4) When data masking is required, you first create a master masked clone from the mirror of your production database (or a Snapshot copy from the appropriate time) and then clone that as many times as necessary.

To create a master masked clone, you simply clone the current database or the Snapshot copy that you want and then use your existing scripts or the Oracle Data Masking Pack to mask out sensitive data. As with the other processes described in this paper, the masking process can be automated using SnapManager for Oracle to decrease the possibility of user error.

Once you've created one or more master masked clones, you then clone the masters so that unmasked user data never appears in test databases. (In situations in which you must replay redo logs to bring a database to a particular point in time it may be necessary to reapply data masking.)

8 CONCLUSION

By following the guidelines set out here, you can create an Oracle environment that is highly resistant to hardware failures, readily supports efficient backup and recovery, provides a mirrored copy for disaster recovery, and supports cloning on demand.

Rapid and efficient cloning of production databases can significantly improve a variety of aspects of your operations:

- **Development and test.** Most shops refresh database copies only once every 90 days. Because making clones has zero impact, you can refresh the cloned production data you use for development work much more often, so that you always test against current, rather than stale, data.

  Instead of having all your developers and testers share one or two copies of a test database, you can create a gold copy and clone it multiple times so that each person has his or her own clones to work with. You can even do destructive testing without affecting anything outside the clone. When the testing is finished, you simply delete the clone and create a new, clean clone image in a matter of minutes. The net result can be significant improvements in development and test capabilities that can lead to improvements in application quality and faster application delivery.
- **Upgrade testing.** Apply major and minor Oracle database upgrades and patches to cloned copies of your production database for thorough testing before implementing these changes.
- **Business intelligence.** The situation is similar for business intelligence reporting. You can quickly create an up-to-date clone of your production database on secondary storage and run your reports against data that is as current as possible. More timely data can translate to better business decisions.
- **Root cause analysis.** You can clone the most recent Snapshot copy made before a crash or other event and then use Oracle tools to replay the log files up to the exact point when the event occurred to analyze the event as it unfolds.
- **DR testing.** You can fully test your DR procedures using cloned copies of DR volumes as often as necessary without disrupting ongoing replication.
- **Data masking**. You can determine that database copies are appropriately masked for their intended purpose using automated processes.

Cloning on demand is a simple and efficient solution that addresses many of the biggest headaches that DBAs face.

You can also use SnapManager for Oracle, which works with NetApp FlexClone and other NetApp technologies, to provide wizard-driven automation of the processes described in this paper.

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

www.netapp.com