

# **EMC CLARiiON Database Storage Solutions: Oracle 10g/11g with CLARiiON Storage Replication Consistency**

*Applied Technology*

---

**Abstract**

This white paper documents how the EMC® CLARiiON® storage replication consistency features of SnapView™ and MirrorView™/Synchronous, together with Oracle's flashback features, facilitate backing up an online Oracle 10g release 2 or Oracle 11g database in UNIX and Windows environments.

August 2008

---

---

Copyright © 2006, 2008 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number H2104.2

---

EMC CLARiiON Database Storage Solutions: Oracle 10g/11g with CLARiiON Storage  
Replication Consistency  
Applied Technology

---

## Table of Contents

|   |           |
|---|-----------|
| <b>Executive summary .....</b>  | <b>4</b>  |
| <b>Introduction .....</b>   | <b>4</b>  |
| Audience .....  | 5         |
| <b>CLARiiON storage systems .....</b>   | <b>5</b>  |
| <b>CLARiiON layered software .....</b>  | <b>6</b>  |
| SnapView overview.....  | 6         |
| SnapView snapshot.....  | 6         |
| SnapView clone.....   | 7         |
| MirrorView/Synchronous (MV/S) overview .....  | 8         |
| SnapView and MirrorView/S consistency .....   | 8         |
| SnapView snapshot consistency.....  | 8         |
| SnapView clone (BCV) consistency.....   | 9         |
| MirrorView/S consistency groups.....  | 9         |
| <b>Application-based consistency and storage-based consistency replication<br/>for Oracle deployments.....</b>        | <b>10</b> |
| Application-based consistency.....  | 10        |
| Storage-based consistency.....  | 11        |
| <b>Leveraging storage-based consistency replication with Oracle Flashback<br/>Technology.....</b>                     | <b>11</b> |
| Oracle Database 11g backup/recovery .....   | 11        |
| Flash Recovery Area.....  | 11        |
| Flashback logs .....  | 12        |
| Flashback Database.....   | 12        |
| Automatic Storage Management .....  | 12        |
| <b>Replicating and recovering an online Oracle 11g database using storage-<br/>based consistency replication.....</b> | <b>13</b> |
| Database file layout .....  | 14        |
| ASM instance parameter file.....  | 14        |
| Database instance parameter file .....  | 15        |
| Replication using SnapView consistency .....  | 15        |
| Using SnapView snapshot consistent session start.....   | 15        |
| Using SnapView clone consistent fracture.....   | 16        |
| Replication using MirrorView/S consistency groups.....  | 18        |
| Creating MirrorView/S consistency groups .....  | 19        |
| Using MirrorView/S consistency groups.....  | 19        |
| Using SnapView with MirrorView/S consistency groups.....  | 21        |
| Recovery using Oracle 11g Flashback Database.....   | 22        |
| <b>Consistency Test Matrix and results.....</b>   | <b>23</b> |
| <b>Conclusion .....</b>   | <b>24</b> |
| <b>References .....</b>   | <b>24</b> |
| Product documents .....   | 24        |
| White papers.....   | 24        |

---

## Executive summary

An Oracle database typically resides on multiple logical units (LUNs) with data that has logical relationships and dependent-write I/Os. To replicate such a database, it is critical that this dependent-write consistency be preserved. This could be accomplished by either shutting down the database or putting the database in hot backup mode before starting the replication process. Either case would adversely impact users of the database in terms of downtime and performance degradation during an online backup. With the consistency features of EMC® CLARiiON® SnapView™ and MirrorView™/Synchronous (MV/S) replication software, it is now possible to replicate an Oracle database without first shutting it down or putting the database in hot backup mode. EMC's proven storage-based software consistency features, together with Oracle's database flashback feature, open up new options for simplifying Oracle database replications.

This white paper describes how the combination of SnapView and MirrorView/Synchronous consistent storage replication technology, Oracle Flash Recovery Area and database flashback features, and the Oracle Automatic Storage Management (ASM) feature facilitates backing up an online Oracle database. Examples of online backups of an Oracle 11g database using EMC replication technology in an ASM environment are presented as well. This paper also replaces the white paper *EMC CLARiiON Database Storage Solutions: Oracle 10g with SnapView in SAN Environments* formerly found on EMC.com and Powerlink.

## Introduction

EMC SnapView snapshots, SnapView clones, and MirrorView/Synchronous are optional CLARiiON storage-system resident software that provides local and remote array-based data replication capabilities. These capabilities range from creating single point-in-time backup copies to creating multiple replicas for disaster protection, all without using host resources. SnapView clones, snapshot images, and MV/S copies can serve as a base for system backups, decision support systems (DSS), revision testing, or in any situation where a consistent, reproducible image of data is needed.

Oracle databases typically span multiple LUNs and, when replicated, the ordering of write I/Os to these LUNs must be maintained. Even though replicating these LUNs takes only a matter of seconds using SnapView and MirrorView, there is still a potential window where the content of one LUN might not be content-consistent with that of another LUN as each LUN was replicated individually. This issue can be addressed operationally by either shutting down the Oracle database or putting the database in hot backup mode prior to starting the replication process. Given the 24/7 uptime requirements generally needed by IT today, hot backup mode is preferable to shutdown mode for Oracle database backup. As long as the Oracle database is kept in hot backup state, Oracle ensures that the content of all files are dependent write-order consistent. For a large database with many files spread across multiple LUNs, the time it takes to put the database into hot backup mode, replicate all the LUNs, then take the database out of hot backup mode can be significant. While the database may be available for reads and writes when in hot backup mode, it does incur host-side performance impact as Oracle has to do more bookkeeping.

However, with the addition of storage-based consistency features for SnapView and MV/S, it is no longer necessary to shut down or put the database in hot backup mode during replication. When consistent replication is performed, any incoming modification to the set of LUNs comprising the database is briefly blocked by the storage system, thus maintaining dependent write-order consistency on the replicated set. This replicated set will be in a state comparable to that of a sudden power failure or crash of the server – it is a coherent Oracle restartable database image. This restartable image, when used in conjunction with Oracle's Flash Recovery Area and Flashback Database features, can be subsequently rolled forward using captured archive logs. Additionally, because SnapView and MV/S operate at the storage system level rather than at the server application level, this model can be used to ensure transactional integrity in a distributed or federated database environment that has write-order dependencies across various applications.

To ensure data integrity and correctness of behavior when replicating an online Oracle database, hot backup tests from the snapshot test kit provided by Oracle as part of its Oracle Storage Compatibility Program (OSCP) were modified and used to test CLARiiON replication consistency. Automatic Storage

---

Management (ASM), a key Oracle database feature since Oracle 10g, was implemented for storing the Oracle database. ASM simplifies the management, placement, and control of Oracle data, thereby lowering the total cost of ownership without compromising performance or availability.

## **Audience**

This white paper is intended for database and systems administrators interested in implementing backup and remote disaster protection plans on UNIX and Windows platforms for Oracle databases using the consistency features of EMC CLARiiON SnapView and MirrorView/S. The reader should be familiar with Oracle database software and ASM and EMC CLARiiON SnapView and MirrorView replication technologies.

## **CLARiiON storage systems**

EMC CLARiiON storage systems are designed for high availability. With redundant components such as dual storage processors (SPs), dual back-end Fibre Channel (FC) loops, dual-ported disk drives, and global hot-spare technology, CLARiiON storage systems do not have any single points of failure. CLARiiON offers RAID 1, 1/0, 3, and 5 options for data protection. These configurations offer protection against single-disk failure. Additionally, as of FLARE<sup>®</sup> release 26, CLARiiON RAID 6<sup>1</sup> technology was introduced. RAID 6 protects against up to two disk failures within a single RAID group, reducing the risk of data loss as a result of double drive failures.

With the dual SPs in CLARiiON systems, performance can be increased by balancing LUNs across SPs. In the event that one SP is unavailable, the other SP will take over servicing I/O requests for the unavailable SP. The SP failure is handled automatically with no noticeable service interruptions. Once the failure is corrected, service automatically fails back to the originally assigned SP if PowerPath<sup>®</sup>'s periodic autorestore feature is enabled; otherwise, a manual restore has to be performed.

The EMC CLARiiON replication methodologies discussed in this paper for use with Oracle extend to all models in the CLARiiON family that support SnapView and MirrorView, including the CX, CX3, and the latest CLARiiON CX4 series storage.

The CLARiiON CX4 series storage platform, launched in Q3 2008, is EMC's state-of-the art midrange family of storage systems. The CX4 features a 64-bit enhanced FLARE driver for increased scaling and operational capability and a new UltraFlex<sup>™</sup> I/O module design that allows additional connection ports to be added to expand connection paths from servers to the CLARiiON. The flexibility in this design enables upgrades to new higher-bandwidth connection technologies, such as 10 Gigabit Ethernet or 8 Gb FC, as they become available. Unmatched scalability is delivered with this platform with up to 960 drives and 4,096 LUNs with the CX4-960 model for accommodating a multitude of drive types ranging from extremely high performance drives to extremely dense drives. The result is not only good scaling in total capacity, but also storage tiering to allow for the most cost-effective manner of storing and managing Oracle database solutions. For additional information on the CLARiiON CX4 UltraScale series, refer to the EMC.com website.

Navisphere<sup>®</sup> Manager and Navisphere CLI, part of the EMC Navisphere family of software products, are management tools used to configure and manage CLARiiON storage systems connected to hosts on a network. Navisphere Manager is a Web-based user interface that lets administrators securely manage CLARiiON storage systems using a common browser. Navisphere CLI is a command line interface that complements or can be used as an alternative to Navisphere Manager. Management functions, such as those for SnapView and MV/S discussed later in this paper, can be automated through shell scripts and batch files using Navisphere CLI commands. Using these management tools, physical disk drives are organized into RAID groups. LUNs of varying sizes can be bound in each RAID group and inherit the RAID characteristics of the group. In turn, LUNs are assigned to storage groups associated with one or

---

<sup>1</sup> The *EMC CLARiiON RAID 6 Technology - A Detailed Review* white paper provides more information on RAID 6.

---

more<sup>2</sup> hosts that are connected to the CLARiiON storage system. The LUNs in a storage group become visible disk devices for each host in the storage group. For further details on creating RAID groups, binding LUNs, and creating storage groups on CLARiiON storage systems, refer to *EMC Navisphere Manager Help 6.X*.

In addition to the Navisphere CLI interface, SnapView snapshots and clones can also be managed using *admsnap*, a server-based utility. The *admsnap* utility can be installed on any server connected to storage systems that have the SnapView software installed and enabled. *Admsnap* cannot be used to set up SnapView; it can only be used to manage ongoing SnapView operations.

## CLARiiON layered software

CLARiiON layered software is optional, storage system-based applications that provide local and remote array-based data replication capabilities. These capabilities range from creating single point-in-time backup copies to creating multiple replicas for disaster protection. The layered applications run on the CLARiiON storage system so no host resources are required to replicate the data. Two of these applications, namely SnapView and MirrorView/Synchronous and their consistency features, will be discussed in this paper.

### ***SnapView overview***

SnapView can be used to create local point-in-time snapshots and full-copy clones of production data for nondisruptive backup. The snapshot images and fractured clones are then available to be mounted on a secondary server to be used for other repurposing such as backups, decision support, or testing. Should primary server access to the production database be interrupted, SnapView snapshot and clone images ensure reliable and quick access to the data from a secondary server. Additionally, data from a snapshot image or clone can be restored back to its source LUN in the event of a data corruption on the source LUN.

### **SnapView snapshot**

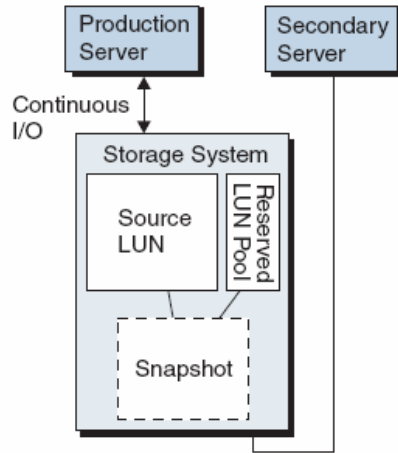
Each SnapView snapshot represents a point-in-time logical image of its source LUN and takes only seconds to create. This point-in-time image of a LUN is captured when a snapshot session is started. A snapshot appears like a normal LUN to secondary servers and can be used for backup, testing, or other repurposing. Snapshots rely on the copy-on-first-write (COFW) technology to track source LUN changes from the time when the snapshot was created. Any writes to the source LUN will result in SnapView copying the original block into a private area on the storage system called the Reserved LUN Pool. This COFW occurs only once for each data block that is modified on the source LUN. Since only changed data blocks are retained in the reserved LUN, the storage capacity required to implement snapshots is a fraction of the size of the source LUN. Because snapshots are virtual point-in-time copies and require access to the unchanged data in the source LUN, any failure affecting its source LUN will make the snapshot useless. If a snapshot is to be used as a database backup, it should be copied to durable media.

Starting with FLARE release 24, all SnapView sessions are persistent by default. A persistent snapshot session provides an added level of protection in that the snapshot continues to be available following SP reboot or failure, storage system reboot or power failure, or peer trespassing. In addition, only a persistent snapshot session can be rolled back. The rollback feature of SnapView is used to replace the contents of the source LUN(s) with the snapshot session's point-in-time data, should the source LUN(s) become corrupt or if a snapshot session's point-in-time data is desired for the source. As soon as the rollback operation is confirmed, the production server can instantly access the snapshot session's point-in-time data while the actual copying of the data back to the source LUN(s) continues in the background. If the snapshot session has been activated and data changes are made to the snapshot, the session will need to be deactivated prior to a rollback if the desired action is to restore the source LUNs to the point in time when the session was started.

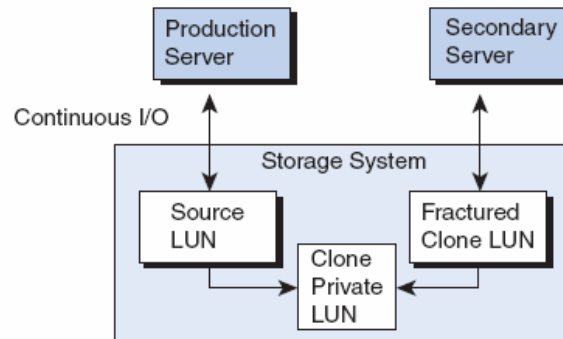
---

<sup>2</sup> Multiple hosts can be connected to the same storage group only for clustered environments.

As shown in Figure 1, a snapshot is a composite of the unchanged data from the source LUN and data that has been saved on the reserved LUN in the Reserved LUN Pool. Because snapshots are both readable and writeable, any snapshot writes from the secondary server are saved on the reserved LUN as well, allowing those changes made to the snapshot to be copied back to the source LUN during a rollback operation with an activated snapshot.



**Figure 1. Snapshot example**



**Figure 2. Clone example**

## SnapView clone

Each SnapView clone is a full bit-for-bit replica of its respective source LUN, requires the exact same disk space as its source, and will initially take time to create. The initial creation time depends on the size of the source LUN to be cloned. Once created, clones are similar to snapshots in that they can be fractured in seconds, thus providing a point-in-time replica of the source LUN that is fully readable and writeable when presented to a secondary server. Unlike snapshots, clones are fully usable even if its source LUN fails. Clones provide users the capability to create fully populated copies of LUNs within a single storage system. While a clone is in a synchronization state with its source LUN, writes to the source LUN are simultaneously copied to the clone as well. To preserve a point-in-time copy, the clone must be fractured from its source LUN. Once fractured and presented to a secondary host, clones are available for I/Os. Changes to either the source or clone LUN are then tracked in the fracture log, a bitmap contained on disk referred to as the Clone Private LUN.

Clones that have been fractured can be made available for other uses such as backups, decision support, or testing. In the event of a data corruption on the source LUNs, the incremental reverse-synchronization feature of SnapView clones can be used to quickly restore their contents to the point in time when the clones were fractured provided that no modifications have been made to the clones; otherwise, the source will reflect the state of the clone at the time of the reverse sync. As an added level of protection, reverse synchronization can be performed in a protected manner by selecting the Protected Restore feature before initiating a reverse synchronization. Because the source LUNs can be brought back online as soon as the reverse synchronization is started for each of the LUNs, selecting Protected Restore prevents any server writes made to the source LUNs from being copied to their clone during the reverse-synchronization process and additionally, the clone that initiated the reverse synchronization is automatically fractured after the reverse synchronization has completed. This feature essentially ensures a “gold” backup copy of the production database that can be used to perform restore operations multiple times from the same set of clones for test or recovery purposes.

Figure 2 illustrates that as the source and fractured clone LUN are changed by their respective server, the Clone Private LUN tracks areas on the source and clone that have changed since the clone was fractured. This logging significantly reduces the time it takes to synchronize or reverse-synchronize a clone and its source LUN because it is incremental; only modified data blocks are copied.

---

## MirrorView/Synchronous (MV/S) overview

MirrorView is a storage system-based disaster recovery (DR) product that provides replication of production data stored in a LUN (primary LUN) on one CLARiiON system to a corresponding LUN (secondary LUN) on a different CLARiiON storage system. MV/S mirrors data synchronously in real time between LUNs on the local and remote storage system. In synchronous mode, each server write to the local (primary) storage system is acknowledged back to the server only after the data has been successfully transferred to the remote (secondary) storage system. Synchronous mode guarantees that the remote image is a complete and exact duplication of the source image.

Figure 3 depicts a sample remote mirror configuration. SnapView can be used in conjunction with MV/S to create a snapshot or clone (FLARE release 24 and later) of the primary or secondary mirror image that can be used to perform verification and run parallel processing processes such as backup, reporting, or testing. This provides an added level of protection at both the local and remote sites should either of these become corrupt.

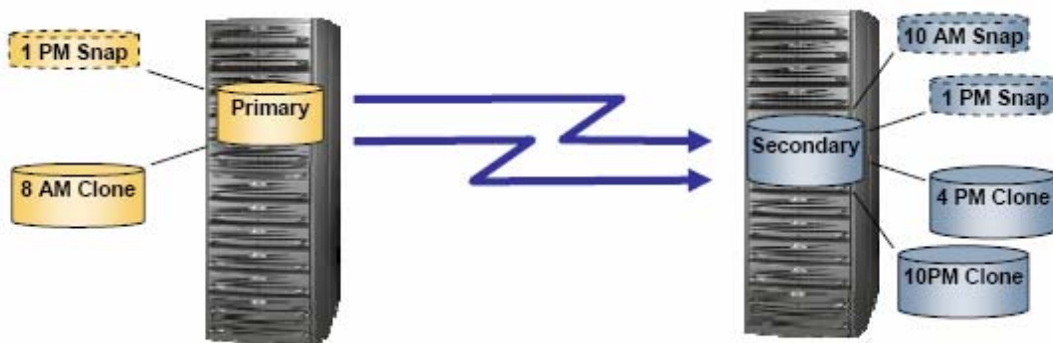


Figure 3. MirrorView/Synchronous example

## SnapView and MirrorView/S consistency

Storage system-based SnapView and MV/S consistency features operate independent of the Oracle application. Consistent replication operates on multiple LUNs as a set such that if the replication action fails for one member in the set, replication for all other members of the set are canceled or stopped. Thus the contents of all replicated LUNs in the set are guaranteed to be identical point-in-time replicas of their source and dependent-write consistency is maintained. This set of LUNs must reside on a single storage system; they cannot span multiple storage systems.

When the consistent replication process is invoked for a set of LUNs comprising the Oracle database, the storage system momentarily holds any writes to each of the source LUNs in the set, long enough for the replication function to complete. With consistent replication, the database does not have to be shut down or put into “hot backup mode.” Replicas created with SnapView or MV/S consistency operations, without first quiescing or halting the application, are restartable point-in-time replicas of the production data and guaranteed to be dependent-write consistent.

### SnapView snapshot consistency

Snapshot consistency is based on the concept of a consistent LUN set. A consistent snapshot session operates on multiple LUNs as a set. This set of LUNs, selected dynamically at the start of each snapshot

---

session, typically contains application-level interrelated contents. Once a consistent session is started, no additional LUNs can be added to that session. In the case of an Oracle database, this would be the set of LUNs containing files with related contents that comprise the database. Any I/O requests to this set of source LUNs is delayed briefly until the session has started on all LUNs within the set, thereby ensuring that a point-in-time dependent write-order consistent restartable copy of the database is replicated. Though stopping a consistent snapshot session on an individual LUN member is allowed, it is highly discouraged because, in most cases, removing a member would render the set incomplete and no longer consistent.

### SnapView clone (BCV) consistency

A SnapView consistent clone fracture is when more than one clone is fractured at the same time in order to capture a point-in-time restartable copy across the set of clones. In the case of an Oracle database, this would be the set of clone LUNs containing files with related contents that comprise the database such as datafiles and log files. This set of clone LUNs is selected dynamically when the fracture is initiated and each clone in the set must belong to different Clone Groups, that is, a consistent fracture cannot be performed on multiple clones belonging to the same source LUN. The SnapView driver will delay any I/O requests to the source LUNs of the selected clones until the fracture has completed on all the clones, thus ensuring that a point-in-time dependent write-order consistent restartable copy of the database is replicated. After the consistent fracture completes, there is no longer any association between the clones in the set. This means subsequent actions such as synchronization, reverse synchronization, or remove are performed on an individual clone basis. To maintain a consistent point-in-time copy of the data among the related clones, it is highly recommended that any action performed on one of the clones in the set should be performed on all of the clones in the set.

### MirrorView/S consistency groups

MV/S includes support for the consistency groups feature. A consistency group is a set of synchronously mirrored pair LUNs that have been identified to have content related and dependent write-order consistent data, such as that of an Oracle database, which must be replicated as a set in order to be usable. Using consistency groupings, MirrorView maintains write ordering across multiple secondary LUNs in the event of an interruption of service to one, some, or all of the write-order dependent volumes. Mirrored LUNs that are members of a consistency group cannot be individually fractured, synchronized, or promoted; these actions have to be performed to all member mirrors of the group as a whole. If a write against a LUN in the group cannot be successfully mirrored to its corresponding LUN in the secondary array, the group will be fractured. A fracture can occur either automatically, due to a mirroring path failure of one or both SPs, or manually. In either case, MirrorView will fracture all members in the consistency group. Any writes that have not been mirrored are not acknowledged to the host until the fracture has completed on all the members of the group. This ensures data integrity is preserved across the set of secondary images.

The contents of the set of fractured images on the secondary site may be somewhat behind that of the production site, but they do represent a point-in-time dependent write-order consistent restartable database. When a consistency group is fractured, any changes to the primary images are tracked in the fracture log for as long as the secondary image is unreachable. The fracture log is a bitmap that is maintained in the storage processor's memory. This bitmap, or fracture log, reduces the time it takes to synchronize the consistency group after it has been fractured.

In the event that the server and/or storage system at the primary site fails, the secondary set of images can be quickly promoted to take over the role of the primary, thus allowing continued access to the production data. After the secondary images have been successfully promoted, additional application recovery procedures may be required to bring the application online at the remote site. In the case of an Oracle database, this would mean restarting the Oracle instance. Because the database was not shut down in an orderly manner, Oracle will have to perform a crash recovery before it can properly open the database.

## Application-based consistency and storage-based consistency replication for Oracle deployments

With the consistency features of SnapView and MV/S, sites currently running Oracle deployments now have the option of selecting either application-based or storage-based consistency when replicating an online Oracle database. The methods are different but both maintain dependent write-order consistency between the relevant LUNs in the replicas. As depicted in Figure 4, Oracle-based consistency creates a valid backup of an Oracle database while storage-based consistency creates a coherent Oracle restartable database. However, this restartable image, when used in conjunction with Oracle 11g Flash Recovery Area and Flashback Database features, can be subsequently rolled forward using captured archive logs.

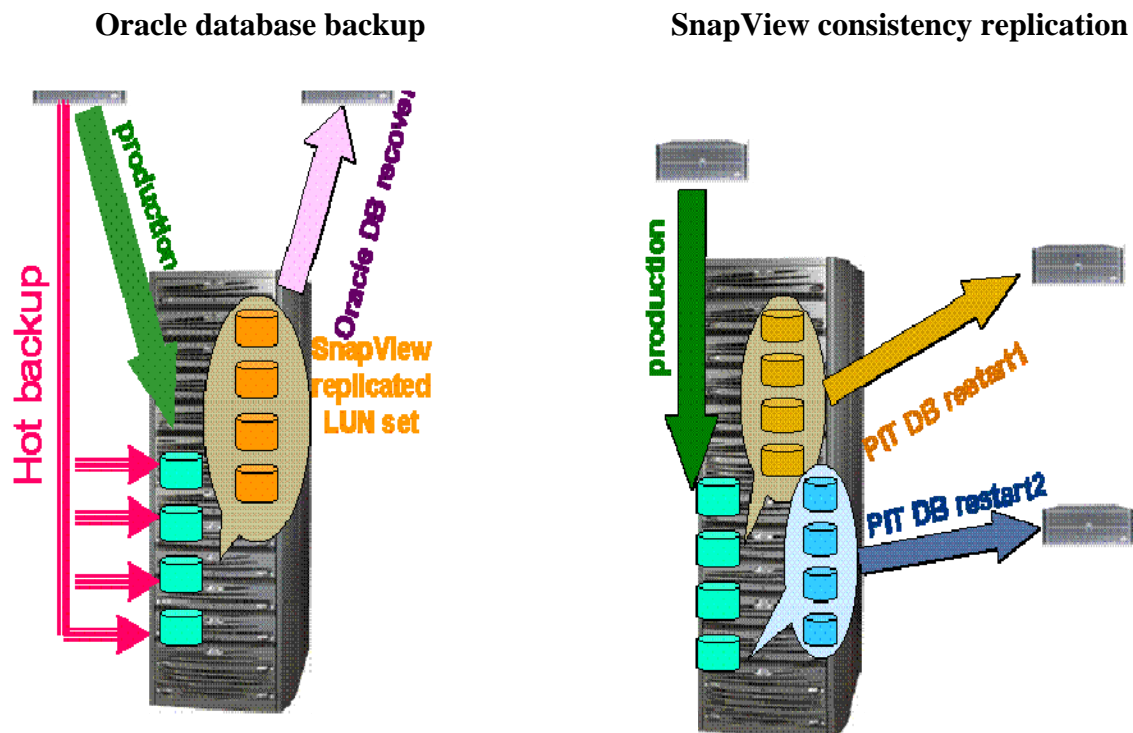


Figure 4. Creating application-based and storage-based database replicas

### Application-based consistency

Prior to the R19 release of the SnapView and MV/S consistency features, both applications have been proven, via extensive testing and Oracle's OSCP program, to provide effective methods for backing up an Oracle database with minimal downtime and minimal performance impact. Without storage-based consistency capabilities, Oracle-based consistency was required to ensure that what the storage system replicates will indeed be a valid Oracle database backup. This means, to take a backup of an online Oracle database, the database has to be put into hot backup mode prior to executing any storage-based replication commands. When an Oracle database is put into hot backup mode, Oracle manages further I/O changes against those LUNs at the Oracle level. While the database is in this quiescence state, all the relevant LUNs can be safely replicated prior to taking Oracle out of hot backup mode and resuming I/O. With this replicated copy, Oracle can restart and recover (recovery model) the database to a coherent point in time after the backup was made. Online backups using Oracle's method of enforcing consistency creates a valid Oracle database backup.

The recovery model is generally more precise because transaction logs can be applied as appropriate against the database to recover the database to a consistent point in time. A consistent database can then be

---

rolled forward to a particular point in time or change number with additional archive log changes. On the other hand, putting a database in hot backup mode will incur host-side performance impact associated with the extra check pointing, logging, and log switching that Oracle has to do as part of this process.

## ***Storage-based consistency***

Because storage-based consistency operates independent of the Oracle application on the server, the database does not have to be put into hot backup mode. Assuming the requisite steps of identifying relevant Oracle database LUNs have been done, the storage system briefly holds any incoming writes against all LUNs in the set so as to create a dependent write-order consistent image of the database. Normal I/O against the LUN set resumes once replication completes. With this replicated copy, Oracle can restart (restart model) the database, but the database cannot be rolled forward. Online backups using only a storage-based method of enforcing consistency create a coherent Oracle restartable database.

The restart model, although less precise, provides a simpler process for resuming operations. The state of the replicated set is comparable to the state of the production database that had crashed due to a sudden power failure or system crash. Thus, the process to restart operations is identical to the process to restart the operation at the original production site after an unexpected interruption. Because the database does not have to be put into hot backup mode, host-side performance is minimally impacted during the replication process.

## **Leveraging storage-based consistency replication with Oracle Flashback Technology**

Oracle Flashback Technology provides a set of features to view and rewind data back and forth in time. Two key features in the area of Oracle Flashback Technology are Flashback Database and Flash Recovery Area. Flashback Database uses flashback logs in the Flash Recovery Area to return the entire Oracle database to a previous point in time. The Flash Recovery Area is a centralized disk location managed and used by Oracle for storing all Oracle database backup and recovery-related files such as control file, online log files, archived redo log files, and flashback logs. Oracle recommends that the Flash Recovery Area be set up using Automatic Storage Management (ASM) because of the benefit of automatic deletion of files no longer needed when space is required for more recent backups.

### ***Oracle Database 11g backup/recovery***

Flashback Database can be used to quickly return a database to an earlier consistent point in time to correct problems caused by logical data corruptions or user errors. Oracle does this by using past block images, captured in Flashback Logs, to back out changes to the database. This feature can be used only if a Flash Recovery Area is configured and the flashback feature enabled.

As stated earlier, using CLARiiON's storage-based consistency to capture an online Oracle database that has not been put into hot backup mode creates only a coherent Oracle restartable database; it is not a valid Oracle backup database. However, with Oracle's Flashback Database feature, this restartable database can be flashed back to a known consistent point in time. Once the database has been flashed back to a consistent state, the appropriate archived logs can now be applied against this database to roll it forward. With this method, CLARiiON's MV/S or SnapView storage-based consistency can be used to generate a coherent Oracle restartable image that archived logs can be applied against to roll it forward without having to put the database into hot backup mode.

### **Flash Recovery Area**

The Flash Recovery Area is an Oracle-managed disk storage location for backup- and recovery-related files such as the control file, archived logs, and flashback logs. Using a Flash Recovery Area simplifies the ongoing administration of an Oracle database. Oracle automatically manages the disk space allocated for

---

recovery files, retaining them as long as they are needed for restore and recovery purposes, and deleting them when they are no longer needed to restore an Oracle database.

The maximum size of the Flash Recovery Area and the retention policy that determines how long backups and archived logs need to be retained for recovery are defined by the user. When the space used in the Flash Recovery Area reaches the specified limit, Oracle automatically deletes the minimum set of existing files from the Flash Recovery Area that are obsolete. Allocating sufficient space to the Flash Recovery Area will ensure faster, simpler, and automatic recovery of the Oracle database. Oracle's recommended disk limit is the sum of the database size, the size of incremental backups, and the size of all archive logs that have not been backed up to tertiary storage. The Flash Recovery Area should also be on a disk separate from the production database files so as to prevent loss of both the production database files and backups in the event of a media failure. Use the SQL statements `DB_RECOVERY_FILE_DEST_SIZE` and `DB_RECOVERY_FILE_DEST` to specify the maximum size and location, respectively, of the Flash Recovery Area.

## Flashback logs

Flashback logs, stored in the Flash Recovery Area, are Oracle-generated logs used to support database flashback operations. The flashback feature must be enabled in order for Oracle to start the collection of database changed pages to flashback logs. Oracle uses these logs to quickly restore the database to a point in the past. To enable logging, issue the SQL statement `ALTER DATABASE FLASHBACK ON`.

## Flashback Database

The Flashback Database process uses flashback logs from the Flash Recovery Area to quickly return an Oracle database to a prior point in time without requiring a backup of the database to first be restored. Flashback Database requires that a Flash Recovery Area be configured. Prior to Oracle 10g release 2, Flashback Database only supported flashback to a prior TIME or SCN. Oracle 10g release 2 added restore points to simplify the recovery process. A restore point is a user-defined name that the database engine uses to map internally to a known SCN that has been committed, thereby eliminating the need to determine the SCN or time of a transaction. This mapping of the restore point name and the SCN is stored in the control file. A normal restore point or a guaranteed restore point can be created at any time using the following SQL command:

```
CREATE RESTORE POINT restore_point [GUARANTEE FLASHBACK DATABASE]
```

Normal restore points eventually age out of the control file if they were not manually deleted. When the flashback logs are aged out of the Flash Recovery Area the aging out is determined by the size of the Flash Recovery Area (`DB_RECOVERY_FILE_DEST_SIZE` database parameter) and the specified retention period (`DB_FLASHBACK_RETENTION_TARGET` database parameter).

Guaranteed restore points on the other hand will never age out of the control file; they must be explicitly dropped. Guaranteed restore points ensure that sufficient flashback logs are always maintained to enable reverting back to that restore point. As a consequence, guaranteed restore points can use considerable space in the Flash Recovery Area. The *Oracle Database Backup and Recovery User's Guide* explains how to size the Flash Recovery Area.

To return the database to a certain restore point, use the name of that restore point in the following SQL command:

```
FLASHBACK DATABASE TO RESTORE POINT restore_point
```

## **Automatic Storage Management**

Automatic Storage Management (ASM) is Oracle's file system and volume manager built specifically for Oracle database files. ASM simplifies database management by consolidating all available storage into ASM disk groups. An ASM disk group is made up of one or more disk devices that ASM manages together as a single logical unit. Instead of having to directly manage potentially thousands of database

---

files, these files can be divided into disk groups, thereby reducing storage management to the disk group level. When creating tablespaces, control files, redo and archive log files, locations of where those files will be placed are specified in terms of disk groups. ASM then manages the file naming and spreads the placement of the database files across all available storage in that disk group. Changes in storage allocation can be adjusted without shutting down the database; disks can be added or dropped from a disk group while the database is running. ASM automatically redistributes (rebalances) the data across all disks in the disk group to ensure an even I/O load and to optimize performance.

An ASM instance, separate from the database instance, is required to manage disks in ASM disk groups. A single ASM instance can service one or more database instances. This ASM instance must be configured and running before the database instance can access any ASM files. As a logical volume manager, the ASM instance has to update ASM metadata that tracks changes about each member disk in the disk group. The ASM metadata itself also resides on the member disk of the disk group. Because the ASM metadata is stored on the same set of member disks as the database files, and as ASM does automatic and dynamic rebalancing, metadata content may be changing even if no user changes are being made to the database content.

When replicating an ASM-managed Oracle database, both the ASM metadata and database data must be in a consistent state during replication in order for the replicas to be usefully repurposed. This means the content of all the member disks of the disk group must not be changing. Currently, there is no specific function in the ASM instance to force a quiescence of an ASM disk group, including its metadata, such that all disk members can be correctly replicated using storage-based replication. The database data can be quiesced by putting the Oracle database into hot backup mode, but there is no easy means to quiesce the ASM metadata. Given the situation, the only way then to reliably perform backups of a database containing ASM files was to use Oracle's Recovery Manager (RMAN) utility.

With the R19 and later support of SnapView and MV/S consistency replication, both ASM metadata and database data can be storage replicated as a point-in-time consistent set. Replicating ASM disk members as a set using SnapView snapshot consistent sessions, SnapView clone set consistent fracture, or MV/S consistency group fracture eliminates the need to quiesce the ASM metadata. As long as the ASM disk members are storage replicated as a point-in-time consistent set, ASM will be able to crash restart the ASM instance and mount the ASM disk group correctly.

## **Replicating and recovering an online Oracle 11g database using storage-based consistency replication**

This section discusses the testing conducted to ensure data integrity and correctness of behavior when replicating an ASM-managed online Oracle 11g database using the consistency features of SnapView snapshots, SnapView clones, and MV/S. Testing was done on a Fibre Channel model CLARiiON CX4-240, but the validation extends to all models in the CLARiiON family that support SnapView and MirrorView. For more detailed information on Oracle 11g and CLARiiON replication software mentioned here, please reference the appropriate documents outlined in the "References" section. In all the test scenarios, the Oracle database was never put into hot backup mode during the consistent replication process. Replications with and without ongoing ASM rebalancing, as well as recovery using Flashback Database, were also covered. All tests were performed on both Linux x86 running RHEL 4.0 update 4 and Windows Server 2003 version 5.2 service pack 1.

Figure 5 is a high-level overview of the steps necessary to replicate and subsequently recover an Oracle 11g database using storage-based consistency replication and the Flashback Database feature of Oracle Flashback Technology.

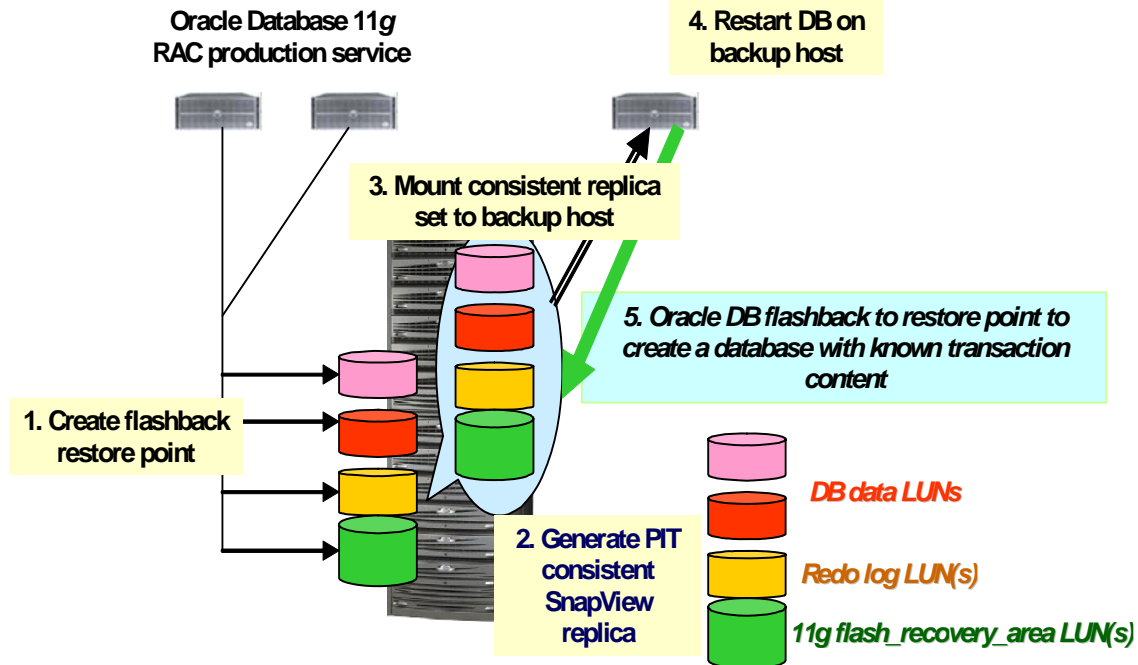


Figure 5. Storage-based consistent replication and recovery

## Database file layout

The ASM-managed Oracle production database files reside on a CX4-240 CLARiiON storage array. In all the test cases, six 4+1 RAID 5 LUNs were created. For SnapView clones, the same number of LUNs were bound on the same storage array and synchronized with its production LUNs. For MirrorView, the production LUNs were mirrored to their corresponding LUNs that were on a separate CX4-240 CLARiiON storage array. The six production LUNs were divided into the following four ASM disk groups that were created with external redundancy specified:

- **DATA\_DGRP** disk group holds all database files and control files. Its member disks (LUNs) are:  
LUN 10 and LUN 11
- **REDO\_DGRP** disk group holds online redo logs. Its member disks (LUNs) are:  
LUN 12 and LUN 13
- **RECOVR\_DGRP** disk group holds flashback logs and multiplexed control files. Its member disk (LUN) is:  
LUN 14
- **ARCH\_DGRP** disk group holds archived redo logs. Its member disk (LUN) is:  
LUN 15

## ASM instance parameter file

An Oracle 11g database using ASM-managed files requires an ASM instance in addition to the regular database instance. Like a regular database instance, an ASM instance has its own initialization parameter file (init\*.ora). Unlike a regular database instance, an ASM instance contains no physical files and has only one required parameter, the `INSTANCE_TYPE = ASM` parameter. This parameter informs Oracle to start an ASM instance and not a database instance. All other ASM relevant parameters have suitable defaults if not set. The following initialization parameters were set in the init\*.ora file for this ASM instance:

```
INSTANCE_TYPE = ASM
ASM_DISKGROUPS = (DATA_DGRP, REDO_DGRP, RECOVR_DGRP, ARCH_DGRP)
```

---

LARGE\_POOL\_SIZE = 12M

## ***Database instance parameter file***

The following initialization parameters relevant to ASM disk groups were set in the init\*.ora file for this database instance:

```
INSTANCE_TYPE = RDBMS
DB_NAME = TestDB
CONTROL_FILES = ('+DATA_DGRP/ctl1TestDB.ctl', '+DATA_DGRP/ctl2TestDB.ctl')
DB_RECOVERY_FILE_DEST_SIZE = 50G
DB_RECOVERY_FILE_DEST = '+RECOVR_DGRP'
LOG_ARCHIVE_DEST_1 = 'LOCATION=+ARCH_DGRP'
```

## ***Replication using SnapView consistency***

The requisite steps necessary to set up a SnapView snapshot or SnapView clone to replicate an Oracle database are the same whether it is for non-consistent or consistent replication. Setup details are provided in the *EMC Navisphere Manager Help 6.X* and *EMC SnapView Command Line Interface (CLI) Reference*. The key difference between non-consistent and consistent replication is in the method the image gets captured. This section will discuss the steps necessary to capture consistent replications of an online database using SnapView snapshot and SnapView clone.

Assume the following in all the snapshot and clone examples:

- The production database files, including archive, redo and flashback logs, are spread across six LUNs configured as ASM-managed files.
- SnapView snapshots, Clone Groups, and clones have been properly created and set up.
- The ASM instance is up.
- The database is running in archivelog mode with Flash Recovery Area and flashback logging enabled.
- The database is currently open with ongoing active transactions from the production server.
- The database is not in hot backup mode during replication.
- LUNs holding the database files, redo log files, and flashback logs will be replicated as a set.
- LUNs holding the archived logs will be replicated as a separate set.

## **Using SnapView snapshot consistent session start**

SnapView snapshot consistent session start is performed by selecting all desired source LUNs and supplying them to the Navisphere CLI “snapview –startsession” command with the “-consistent” switch specified. This replicated set is a coherent Oracle database restartable image. In order to roll this restartable image forward using captured archive logs, some preliminary steps need to be completed prior to and after starting the snap session. This involves creating a “restore point” before starting the session and ensuring active redo logs are archived and captured after the session has been started, all from the production server.

1. Create a new flashback restore point:

```
sqlplus /nolog
SQL> connect sys/manager as sysdba
SQL> drop restore point at3pm
SQL> create restore point at3pm;
```

This creates a normal restore point named “at3pm”, which is an alias for the SCN of the database at that time. As stated earlier, normal restore points will age out depending on the size of the Flash Recovery Area and the specified retention period (default is 1440 minutes). To ensure that the flashback logs for the named restore point will not age out, create a guarantee restore point using the following SQL command instead:

---

```
SQL> create restore point at3pm guarantee flashback database;
```

2. Start a SnapView snapshot consistent session of the LUNs holding the database files, redo log files, and flashback logs:

```
naviseccli -h primary_array snapview -startsession sessionname  
-lun luns -consistent
```

Example:

```
naviseccli -h CX4201a snapview -startsession 3pmDataSession -lun  
10 11 12 13 14 -consistent
```

This Navisphere CLI command starts a consistent session named `3pmDataSession` on LUNs 10, 11, 12, 13, and 14. These LUNs are member disks of ASM disk groups `DATA_DGRP`, `REDO_DGRP`, and `RECOVR_DGRP`. This command takes only seconds to complete. Once completed, a consistent point-in-time restartable image of the database has been captured and can be made available for use on a secondary server.

3. Archive all unarchived logs:

```
sqlplus /nolog  
SQL> connect / as sysdba  
SQL> alter system archive log current;  
SQL> select 'NextChange', next_change# from v$log_history where recid=  
        (select max(recid) from v$log_history);  
SQL> alter database backup controlfile to trace resetlogs;  
  
NEXTCHANGE    NEXT_CHANGE#  
-----  
NextChange           70760
```

All active redo logs are archived in ASM disk group `ARCH_DGRP`. These archived logs are required to recover the point-in-time image of the database captured in step 2.

4. Start a SnapView snapshot consistent session of the LUNs holding the archived logs:

```
naviseccli -h primary_array snapview -startsession sessionname  
-lun luns -consistent
```

Example:

```
naviseccli -h CX4201a snapview -startsession 3pmArchSession -lun  
15 -consistent
```

This Navisphere CLI command starts a consistent session named `3pmArchSession` on LUN 15. This LUN is in ASM disk group `ARCH_DGRP`. With this replicated LUN containing the archived logs, Oracle's database flashback feature can be leveraged to flash the restarted database back to a known SCN as captured in the "at3pm" restore point in step 1, and then roll forward using the archived logs.

At this point, all the necessary files needed to generate a usable valid Oracle backup that logs can be played against have been captured. The section "Recovery using Oracle 11g Flashback Database" discusses the process whereby this replicated set of database LUNs can be recovered and used to generate a valid Oracle backup.

## Using SnapView clone consistent fracture

SnapView consistent clone fracture is performed by identifying all clone LUNs that need to be fractured as a set and supplying them to the Navisphere CLI "snapview -consistentfractureclones" command, thus preserving the point-in-time restartable copy across these clones. This set of clones must belong to a different Clone Group on the same storage system. This replicated clone set is a

---

coherent Oracle database restartable image. In order to be able to roll this restartable image forward using captured archive logs, some preliminary steps need to be completed prior to and after fracturing the clone set. This involves creating a “restore point” before the fracture and ensuring active redo logs are archived and captured after the fracture has successfully completed, all from the production server.

1. Create a new flashback restore point:

```
sqlplus /nolog
SQL> connect sys/manager as sysdba
SQL> drop restore point at3pm;
SQL> create restore point at3pm;
```

This creates a normal restore point named “at3pm”, which is an alias for the SCN of the database at that time. As stated earlier, normal restore points will age out depending on the size of the Flash Recovery Area and the specified retention period (default is 1440 minutes). To ensure that the flashback logs for the named restore point will not age out, create a guarantee restore point using the following SQL command instead:

```
SQL> create restore point at3pm guarantee flashback database;
```

2. Verify that each clone LUN containing the database files, redo log files, and flashback logs is in a Synchronized or Consistent state:

```
naviseccli -h primary_array snapview -listclone -name
clone_groupname -cloneid id -CloneState
```

Example:

```
naviseccli -h CX4201a snapview -listclone -name Data1CGroup
-cloneid 0100000000000000 -CloneState
```

```
naviseccli -h CX4201a snapview -listclone -name Redo1CGroup
-cloneid 0100000000000000 -CloneState
```

```
naviseccli -h CX4201a snapview -listclone -name RecovrCGroup
-cloneid 0100000000000000 -CloneState
```

3. Once it is determined from the previous step that the clones are either in a Synchronized or Consistent state, then initiate a single SnapView consistent clone fracture of this set of clone LUNs:

```
naviseccli -h primary_array snapview -consistentfractureclones
-CloneGroupNameCloneID name1 cloneId1 ... nameN cloneIdN -o
```

Example:

```
naviseccli -h CX4201a snapview -consistentfractureclones
-CloneGroupNameCloneID Data1CGroup 0100000000000000
Data2CGroup 0100000000000000 Redo1CGroup 0100000000000000
Redo2CGroup 0100000000000000 RecovrCGroup 0100000000000000 -o
```

This Navisphere CLI command fractures clone LUNs with a clone ID of 0100000000000000 from Clone Groups Data1CGroup, Data2CGroup, Redo1CGroup, Redo2CGroup, and RecovrCGroup. These LUNs are clones of source LUNs 10, 11, 12, 13, and 14, respectively (member disks of ASM disk groups DATA\_DGRP, REDO\_DGRP, and RECOVER\_DGRP). Once fractured, the set of selected clones is a consistent point-in-time restartable image of the database and can be made available for use on a secondary server.

4. Archive all unarchived logs:

```
sqlplus /nolog
SQL> connect / as sysdba
SQL> alter system archive log current;
```

---

```

SQL> select 'NextChange', next_change# from v$log_history where recid=
      (select max(recid) from v$log_history);
SQL> alter database backup controlfile to trace resetlogs;

NEXTCHANGE    NEXT_CHANGE#
-----
NextChange          81260

```

All active redo logs are archived in ASM disk group ARCH\_DGRP. These archived logs are required to recover the point-in-time image of the database captured in step 3.

5. Verify that the clone LUN holding the archived logs is in a Synchronized or Consistent state (see step 2), and then initiate a SnapView fracture of this clone LUN. Please note that if the archived logs were spread over multiple LUNs, a consistent clone fracture using the `consistentfractureclones` command will have to be initiated as detailed in step 3:

```

naviseccli -h primary_array snapview -fractureclones -name
clone_groupname -cloneid id -o

```

Example:

```

naviseccli -h CX4201b snapview -fractureclone -name ArchCGroup
-cloneid 0100000000000000 -o

```

This Navisphere CLI command fractures the clone LUN with a clone ID of 0100000000000000 from Clone Group ArchCGroup. This LUN is a clone of source LUN 15 (member disk of ASM disk group ARCH\_DGRP). With this replicated LUN containing the archived logs, Oracle's database flashback feature can be leveraged to flash the restarted database back to the known SCN as captured in the "at3pm" restore point in step 1, and then roll forward using the archived logs.

At this point, all the necessary files needed to generate a usable valid Oracle backup that logs can be played against have been captured. The section "Recovery using Oracle 11g Flashback Database" discusses the process whereby this replicated set of database LUNs can be recovered and used to generate a valid Oracle backup.

## Replication using MirrorView/S consistency groups

To capture mirrored images of an Oracle database spread across multiple LUNs such that dependent write-order consistency is maintained requires the consistency group feature of MirrorView. A consistency group is a set of mirrors managed as a single entity that must remain consistent with respect to each other at all times. In addition to the requisite steps necessary to set up primary and secondary images for MirrorView to mirror an Oracle database to a separate storage system, consistency groups require the creation and setup of a consistency group. This section will discuss the steps necessary to set up consistency groups, using MirrorView/S consistency group fracture to facilitate consistent replications of an online Oracle database, and then using SnapView on the secondary storage to create local replicas of the secondary images. For further details on setting up mirror images and consistency groups, refer to *EMC Navisphere Manager Help 6.X* and the *EMC MirrorView/Synchronous Command Line Interface (CLI) Reference*.

Assume the following in all MV/S examples:

- The production database files, including archive, redo and flashback logs, are spread across six LUNs configured as ASM-managed files.
- MirrorView connections between storage systems have been established.
- Remote mirrors with secondary images (LUNs 20, 21, 22, 23, 24, and 25) have been properly created and set up.
- The ASM instance is up.
- The database is running in archive log mode with Flash Recovery Area and flashback logging enabled.
- The database is currently open with ongoing active transactions from the production server.

- The database is not in hot backup mode during replication.
- LUNs holding the database files, redo log files, and flashback logs will be in one consistency group.
- LUNs holding the archived logs will be in a separate consistency group.

## Creating MirrorView/S consistency groups

As in a SnapView snapshot start session and consistent clone fracture, the first step to creating a consistency group is to determine the database source LUNs that must be mirrored as a set in order to maintain write-order consistency among the relevant LUNs. This in turn determines the number of consistency groups to create. The following example creates two consistency groups, one for LUN(s) holding the database files, redo log files, and flashback logs, and the other for LUN(s) holding the archived logs.

1. From the production server, create two MV/S consistency groups on the production (primary) storage system:

```
naviseccli -h primary_array mirror -sync -creategroup -name consistency_groupname -o
```

Example:

```
naviseccli -h CX4201a mirror -sync -creategroup -name mirrorDB_CGroup -o  
naviseccli -h CX4201b mirror -sync -creategroup -name mirrorARCH_CGroup -o
```

This Navisphere CLI command creates two consistency groups named `mirrorDB_CGroup` and `mirrorARCH_CGroup` on the primary storage system.

2. From the production server, add mirrors to the previously created consistency groups:

```
naviseccli -h primary_array mirror -sync -addtogroup -name consistency_groupname -mirrorname mirror_name
```

Example:

```
naviseccli -h CX4201a mirror -sync -addtogroup -name mirrorDB_CGroup -mirrorname Data1_mirror  
naviseccli -h CX4201b mirror -sync -addtogroup -name mirrorARCH_CGroup -mirrorname Arch_mirror
```

The `-addtogroup` command can only add one remote mirror at a time to a consistency group. Repeat this `-addtogroup` command such that remote mirrors `Data1_mirror`, `Data2_mirror`, `Redo1_mirror`, `Redo2_mirror` and `Flash_mirror` are all members of consistency group `mirrorDB_CGroup`.

When a consistency group is created and one or more mirrors are added to it, MV/S automatically creates a consistency group with the same name and content on the secondary storage system. Once mirrors are in a consistency group, they cannot be managed individually; they have to be managed as a group using only consistency group commands.

## Using MirrorView/S consistency groups

MirrorView/S consistency group fracture is accomplished by specifying the name of the consistency group to the Navisphere CLI “`mirror -sync -fracturegroup`” command. The consistency groups should be in either a synchronized or consistent state prior to the fracture. Once fractured, its corresponding consistency group on the secondary storage system can be promoted, snapped, or clone fractured so as to enable I/O access to the secondary image. This set of mirrors on the secondary storage system is a coherent Oracle database restartable image. In order to be able to roll this restartable image forward using captured archive logs, some preliminary steps need to be completed prior to and after fracturing the consistency group. This

---

involves creating a “restore point” before the fracture and ensuring active redo logs are archived and captured after the fracture has successfully completed, all from the production server.

1. Create a new flashback restore point:

```
sqlplus /nolog
SQL> connect sys/manager as sysdba
SQL> drop restore point at3pm;
SQL> create restore point at3pm;
```

This creates a normal restore point named “at3pm”, which is an alias for the SCN of the database at that time. Normal restore points will age out depending on the size of the Flash Recovery Area and the specified retention period (default is 1440 minutes). To ensure that the flashback logs for the named restore point will not age out, create a guarantee restore point using the following SQL command instead:

```
SQL> create restore point at3pm guarantee flashback database;
```

2. Verify that the consistency group that contains the database files, redo log files, and flashback logs is in a Synchronized or Consistent state:

```
naviseccli -h primary_array mirror -sync -listgroups -name
consistency_groupname -state
```

Example:

```
naviseccli -h CX4201a mirror -sync -listgroups -name
mirrorDB_CGroup -state
```

3. Once it is determined from the previous step that the consistency group is in either a Synchronized or Consistent state, then initiate a consistency group fracture:

```
naviseccli -h primary_array mirror -sync -fracturegroup -name
consistency_groupname -o
```

Example:

```
naviseccli -h CX4201a mirror -sync -fracturegroup -name
mirrorDB_CGroup -o
```

This Navisphere CLI command fractures all the mirror images in the consistency group mirrorDB\_CGroup. Members of this consistency group are source LUNs 10, 11, 12, 13 and 14 and their corresponding mirrored LUNs on the secondary storage system (member disks of ASM disk groups DATA\_DGRP, REDO\_DGRP, and RECOVR\_DGRP). Once fractured, its corresponding images on the secondary storage are consistent point-in-time restartable images of the database and can be made available for use on a secondary server.

4. Archive all unarchived logs:

```
sqlplus /nolog
SQL> connect / as sysdba
SQL> alter system archive log current;
SQL> select 'NextChange', next_change# from v$log_history where recid=
(select max(recid) from v$log_history);
SQL> alter database backup controlfile to trace resetlogs;

NEXTCHANGE    NEXT_CHANGE#
-----
NextChange          81750
```

All active redo logs are archived in ASM disk group ARCH\_DGRP. These archived logs are required to recover the point-in-time image of the database captured in step 3.

- 
5. Verify that the consistency group holding the archived logs is in a Synchronized or Consistent state (see step 2), then initiate a consistency group fracture:

```
naviseccli -h primary_array mirror -sync -fracturegroup -name consistency_groupname -o
```

Example:

```
naviseccli -h CX4201b mirror -sync -fracturegroup -name mirrorARCH_CGroup -o
```

This Navisphere CLI command fractures all the mirror images in the consistency group `mirrorARCH_CGroup`. Members of this consistency group are source LUN 15 and its corresponding mirrored LUN on the secondary storage system (member disk of ASM disk group `ARCH_DGRP`). With the secondary mirror images in this consistency group containing the archived logs, Oracle's database flashback feature can be leveraged, from the secondary server, to flash the restarted database back to the known SCN as captured in the "*at3pm*" restore point in step 1, and then roll forward using the archived logs.

At this point, all the necessary files needed to generate a usable valid Oracle backup that logs can be played against have been captured on the secondary storage array. SnapView snapshots or clones can now be used to quickly create replicas of the secondary mirror images on the secondary storage array. Both offers an extra level of protection but clones offer added disk protection and have less of a performance impact than snapshots. Once the SnapView operations completes, the mirroring relationship can be reestablished.

## Using SnapView with MirrorView/S consistency groups

SnapView snapshot and SnapView clone, when used in conjunction with MirrorView/S on the secondary storage, provide local replicas of secondary images. This allows a secondary server access to data at the secondary storage without having to promote the secondary images. SnapView consistent operations cannot be performed on the consistency group level; it has to be performed on the individual members of the consistency group.

Start SnapView snapshot consistent sessions of the secondary mirrored images as follows:

1. From the secondary server, start a consistent snapshot session of the LUNs holding the database files, redo log files, and flashback logs:

```
naviseccli -h CX4202a snapview -startsession 4pmDataSession -lun 20 21 22 23 24 -consistent
```

2. From the secondary server, start a consistent snapshot session of the LUNs holding the archived logs:

```
naviseccli -h CX4202b snapview -startsession 4pmArchSession -lun 25 -consistent
```

Start SnapView clone consistent fracture of the secondary mirrored images as follows:

1. From the secondary server, initiate a consistent clone fracture of the LUNs containing the database files, redo log files, and flashback logs:

```
naviseccli -h CX4202a snapview -consistentfractureclones -CloneGroupNameCloneID Data1CGroup 0100000000000000 Data2CGroup 0100000000000000 Redo1CGroup 0100000000000000 Redo2CGroup 0100000000000000 RecovrCGroup 0100000000000000 -o
```

2. From the secondary server, start a consistent session of the LUNs holding the archived logs:

```
naviseccli -h CX4202b snapview -fractureclone -name ArchCGroup -cloneid 0100000000000000 -o
```

---

The next section, “Recovery using Oracle 11g Flashback Database,” discusses the process whereby this replicated set of database LUNs can be recovered and used to generate a valid Oracle backup.

## **Recovery using Oracle 11g Flashback Database**

In order for a secondary server to access a snapshot or fractured clone, they must belong to a storage group that is connected to the secondary server. The snapshot must then be activated to a SnapView session and mounted on the secondary server. Fracture clones just need to be mounted on the secondary server. For further details on storage group setup and snapshot activation, refer to the *EMC Navisphere Manager Help 6.X*. This section will discuss the steps necessary to mount, start up, and recover the database from the replicated set of LUNs captured using storage-based consistent features.

Whether the replicated set of LUNs was captured using SnapView snapshots, SnapView clones, or MirrorView/S consistency groups, the Oracle startup and recovery process is the same. The following assumptions are made:

- The secondary server is of the same OS and running the same version of Oracle 11g as the production server.
- The replicated set of LUNs has been made accessible to the secondary server.
- An ASM instance has been started on the secondary server.

The storage replicated set of LUNs are bit-for-bit identical to their respective source LUNs. They have the same ASM disk signature information, and thus the same ASM disk groups. Once these LUNs are made accessible to the secondary server, they have to be mounted by an ASM instance using the following sample SQL commands:

```
$sqlplus /nolog  
SQL> connect / as sysdba  
SQL> alter diskgroup DATA_DGRP mount;  
SQL> alter diskgroup REDO_DGRP mount;  
SQL> alter diskgroup RECOVR_DGRP mount;  
SQL> alter diskgroup ARCH_DGRP mount;
```

When the ASM instance mounts the replicated set of LUNs, it logically sees the ASM disk groups as being left open from last use. ASM would then perform the necessary steps to recover transient ASM metadata changes. Once the disk groups are mounted, the Oracle database instance can be restarted and the database flashed back to a restore point captured prior to the LUNs being replicated.

```
$sqlplus /nolog  
SQL> connect / as sysdba  
SQL> startup mount;  
SQL> flashback database to restore point at3pm;
```

When the flashback command successfully completes, the database is left mounted and recovered to the specified restore point. To verify that the database was returned to the desired point in time, open the database in read-only mode and perform some queries to inspect the database contents.

At this point, the database can be recovered and rolled forward by utilizing the set of archived logs captured during the replication process and indicating how far the database should be advanced (by change number or to a particular point in time). The Oracle database can then be made available for updates. The following sample SQL “recover” command recovers the database until the change number as captured in step 3 of the section “Using SnapView snapshot consistent session start”:

```
SQL> recover automatic database until change 70760 using backup controlfile;  
SQL> shutdown  
SQL> startup mount;  
SQL> alter database open resetlogs;
```

The replicated Oracle database is now a fully usable database, suitable for repurposing or be backed up to durable media.

## Consistency Test Matrix and results

| TEST   | RESULTS  |  |  | No. of runs |
|--|--|--|--|-------------|
|  | Start snap session                                     | Clone fracture   | MV/S fracture  |             |
| Flashback <b>ON</b><br>Database <b>NOT</b> in <b>HOT BACKUP</b>                                      | Successfully restarted, flashed back, & rolled forward | Successfully restarted, flashed back, & rolled forward | Successfully restarted, flashed back, & rolled forward | 5           |
| Flashback <b>ON</b><br>Database <b>NOT</b> in <b>HOT BACKUP</b><br>Disk group rebalance in progress  | Successfully restarted, flashed back, & rolled forward | Successfully restarted, flashed back, & rolled forward | Successfully restarted, flashed back, & rolled forward | 5           |
| Flashback <b>OFF</b><br>Database <b>NOT</b> in <b>HOT BACKUP</b><br>Disk group rebalance in progress | Successfully restarted                                 | Successfully restarted                                 | Successfully restarted                                 | 5           |

---

## Conclusion

The consistency features of SnapView and MirrorView/S simplify Oracle deployments requiring database replications. Because the Oracle database no longer needs to be put into hot backup mode during the replication process, impact on host-side performance is minimized. This makes it operationally viable to create usable database replicas more frequently. Replicating an Oracle database using ASM-managed files is also simplified. When ASM disk members are replicated as a set using the consistency features of SnapView or MirrorView, the metadata that ASM depends on for restarting is also replicated. There is no need to manually ensure ASM metadata consistency. ASM will be able to crash restart the ASM instance and remount the ASM disk group. CLARiiON storage-based consistent replications of an Oracle database are guaranteed to be point-in-time replicas of their source and write-order consistent. This replicated set is a coherent Oracle restartable image, and when used in conjunction with Oracle 11g Flash Recovery Area and Flashback Database features, Oracle can recover and roll the database forward using captured archived logs. Combining storage-based consistent replication and the Oracle 11g flashback feature provides the means to create a usable valid Oracle backup without production impact.

## References

### **Product documents**

- *Oracle Database Backup and Recovery User's Guide 11g Release 1 (11.1)*
- *Oracle Database Concepts 11g Release 1 (11.1)*
- *EMC Navisphere Manager Help 6.X*
- *EMC SnapView Command Line Interfaces (CLI) Reference*
- *EMC MirrorView/Synchronous Command Line Interface (CLI) Reference*

### **White papers**

- *EMC CLARiiON SnapView and MirrorView for Oracle Database 10g Automatic Storage Management – Best Practices Planning*
- *MirrorView Knowledgebook - Applied Technology*
- *EMC CLARiiON RAID 6 Technology - A Detailed Review*