Guidelines for Using Snapshot Storage Systems for Oracle Databases

Nabil Osorio and Bill Lee, Oracle Corporation

October 2001

ORACLE
Enabling the Information Age™

Change Record

Date	Author	Version	Change Reference
28-Aug-00	Nabil A. Osorio	1	No previous document
01-Oct-01	Nabil A. Osorio		revise Section 5 to include Async mode

Guidelines for Using Snapshot Storage Systems for Oracle Databases

Introduction

With the constant increase for corporations and organizations to store high volume amount of data, and the need for implementing high available systems (HA) with minimum down time for maintenance, storage system vendors have developed and implemented new technologies to fulfill this requirement. One such technology is snapshot storage systems. Storage snapshot technology may allow Oracle customers to quickly backup and restore large amounts of data without jeopardizing data integrity nor performance of the online service.

This white paper describes the usage of storage snapshot technology with Oracle databases. It shows how local snapshots, which include both ***snapshot of mirror*** and ***copy on write*** techniques**,** and the remote snapshot, which uses ***copy on write*** technique, can be used for hot backups, cold backups, remote point-in-time copy and disaster recovery.

The intended audience are (1) advanced database administrators planning to buy or use snapshot storage systems for Oracle databases, and (2) designers of local and remote snapshot storage systems. This paper assumes that the reader has a full understanding on database backup and recovery methodology for oracle databases and the issues behind each technique. Additionally, a basic knowledge of snapshot storage systems is required.

Technology Description

A storage snapshot is basically a copy image of storage devices or file systems on a secondary local or remote systems at a particular point in time. Snapshots duplicate data from standard storage devices by using the ***snapshot of mirror*** techniques or ***copy on write*** techniques. *Figure i* depicts how Snapshot technique is composed of different categories.

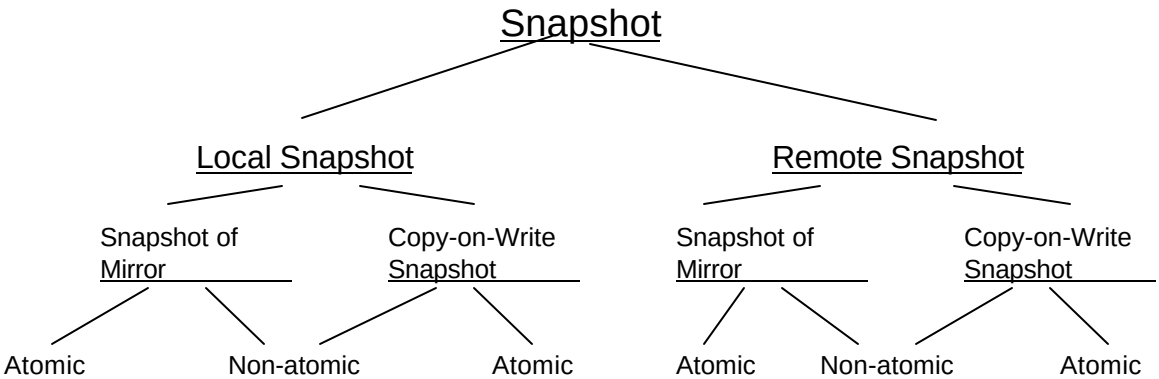

Figure i

Local vs. ***Remote*** refers to where the duplicated images will be stored. In the case of Local, the duplicated image — a snapshot — is at the local site. Some storage systems can let a second host on the local site mount and access the snapshot. In the case of remote, the duplicated image can potentially be at a remote site, accessible by a remote host. Such remote mirroring or snapshot systems are typically designed for disaster recovery. Refer to *Guidelines for Using Remote Mirroring Storage Systems for Oracle Databases* for additional information on Remote Mirroring techniques.

Snapshot of mirror refers to the capability to produce a snapshot disk image by breaking the mirror of a mirroring storage system. The mirroring storage system may be a local or remote mirroring storage system. The break of mirror may be Atomic or non-Atomic.

Copy on Write refers to a technique that produces a snapshot storage image logically, using copy on write techniques. When a user takes a snapshot, the system simply records such an event; it does not physically copy and duplicate all the data at the time the snapshot is taken. As blocks are changed, prior to the change, a before image copy of the block is stored in the snapshot area. While the primary file system continues to change, the snapshot accumulates the original image copy of the data until the next snapshot is generated. This method can also be Atomic or non-Atomic.

Atomic vs Non-Atomic Atomic refers to a total split of the mirrors at one point in time. The real concern is write order, the second physical I/O should not be written before the previous physical I/O. When a link is split, all links must be split at the same time or an ordering mechanism must be in place to assure proper I/O ordering. In other words, ***Atomic*** maintains an ordering write i/o when the snapshot is taken, while ***Non-Atomic*** does not maintain such write ordering when the snapshot is taken. A few usages described in this paper requires the snapshot systems to be atomic. If not sure, users of snapshot storage system should consult the storage vendor as to whether a snapshot storage system is atomic.

For the ***Remote*** capability, propagating data can be accomplished using ***Asynchronous*** (Async) or ***Synchronous*** (Sync) mode, depending on the level of reliability and availability desired. For Sync mode, a write from a local host is acknowledged only until the write has been propagated to the remote or secondary facility. For Async mode writes, the local host acknowledges the write request without verification that the write took place at the remote facility. There is a risk of data loss using the Async mode since writes cannot be verified, and thus the techniques for recovering a database are different. Refer to *Guidelines for Using Remote Mirroring Storage Systems for Oracle Databases* for additional information on Remote Mirroring techniques.

Some storage systems allow a second host to mount and access a snapshot image. Some don't. In the former case, some storage systems may restrict the second host to only read the snapshot image. Some storage system may allow the second host to write in the snapshot area. In the latter case, many storage systems have the capability of removing all the changes made by the second host.

There are many products and vendors in the market that implement the snapshot technology, but the description of those products is beyond the scope of this paper.

Overview of Usage's

This section provides guidelines on how to use Snapshot technology to take a cold database backup, online database backup (also referred as hot backups),

Database Point-in-time image restore for Decision Support System (DSS) reporting capability using remote systems, and disaster recovery.

Another implicit use for snapshots is database point-in-time-recovery which can be accomplished by using hot backups or cold backups. For more details on how to recover a database refer to standard Oracle documentation.

1. Local Cold Backup

Whether using the ***Snapshot of Mirrors*** method or the ***copy on write*** method, cold backups can be implemented by using the standard procedures. All data files, and the initSID.ora files should be backed up. It is also assumed that the control file is being backed up by using the ***alter database backup controlfile to 'filespec'*** statement. For information on how to backup the control file and its use, consult the standard Oracle documentation.

One can use the snapshot mechanism to take a cold backup of an Oracle database quickly. To do this, all the data files to be backed up must be in the volume or file system capable of taking snapshots. Use the following steps:

- Shutdown local database (only Normal or Immediate)
- Split the local mirror (if using the "snapshot of mirrors" method) or take a new storage snapshot (if using "copy-on-write method) set.
- Startup production database
- Resume normal operations

The snapshot is now a cold backup of all the data files.

2. Local Hot Backup

A Hot Backup is a backup taken while a database is open. In this case we are using the hot backup mode to take a snapshot of the database, and thus avoiding the concern of having fuzzy files (a data file that contains at least one block with an SCN more recent than the checkpoint SCN stored in its header). All the data files to be backed up must be in the volume or file system capable of taking snapshots.

The following steps can be used to take on-line backups:

- Alter the tablespaces to begin backup
- Split the local mirror (if using the "snapshot of mirrors" method) or take a new snapshot set (using "copy-on-write method).
- alter the tablespaces to end the backup.
- Resume database operations

The snapshot is now a hot backup of all the data files.

3. Local Point-In-Time (PITI) Image

A point-in-time image is a snapshot taken while an Oracle database is running, without putting all the related tablespaces into hot backup mode. Although local Point-In-Time image restore is achievable, Oracle does not support nor recommend its use. The reason for not using PITI on local configurations is that there is a greater risk to damage the production database if not used correctly: mixing the archive logs before and after the restore can potentially lead to data corruption. Oracle does not provide technical support for this configuration. Nevertheless, PITI can be used on remote configurations where the benefits are much greater and the risk is lower than in local environments. A supported configuration would be with remote hosts. For remote PITI refer to Section 8.

A remote mirroring system is typically designed for disaster recovery. To be able to recover from disasters, the system must always atomically break the mirror when a disaster occurs. For a full description of the usage of remote mirroring storage systems, see *Guidelines for Using Remote Mirroring Storage Systems for Oracle Databases.* Some of the usages describe below applies to remote snapshots of mirror.

For Remote snapshot copy-on-write to work, it can be atomic or non-atomic. Conceptually, such system can push (sending image copies to remote host) snapshots to a remote host. Ask product manufacturer for a list of capabilities of a particular snapshot system. Such systems can also be used for a limited form of disaster recovery that can potentially lose more data than remote mirroring storage systems. Section 9, covers disaster recovery fail-over and fallback procedures using such storage systems.

4. Remote Cold backup

Similar to section **1. Local cold backup**, both methods of snapshots ***(Snapshot of Mirror & Copy on Write***) can also be used for cold backup:

The steps to take are as follows:

- Shutdown local database (Normal or Immediate)
- Propagate all changes to secondary site (if Async mode)
- Split the remote mirror (if using the “snapshot of mirrors” method) or take a new storage snapshot set (if using “copy-on-write method).
- Startup primary production database
- Resume operations at primary site
- Use the remote secondary database to take the cold backup at any time.

The remote snapshot is now a cold backup of all the data files.

5. Remote Hot backup

A Hot Backup on a local host is used to take a copy of the data files which can be used later in recovery. A Hot Backup on a second host can also be used to backup data files, as well as start a new instance. An Instance can be started from a snapshot image. Similar to the Hot Backup on a local host, this method

avoids the concern of fuzzy files. Again the steps are the same for a Hot Backup whether local or remote.

To take a remote hot backup use the following steps:

- Alter the tablespaces to begin backup
- Propagate all changes to secondary site (if Async mode only)
- Split the remote mirror (if using the “snapshot of mirrors” method) or take a new snapshot set (using “copy-on-write method).
- alter the tablespaces to end the backup.
- Use the remote secondary database to take the backup at any time

The remote snapshot is now a hot backup of all the data files.

A snapshot image may be accessible to a second host in two ways. (1) The storage system may support local snapshots, and it allows a second local host to mount and access the snapshot image. (2) the storage system may support remote snapshots, either because it is a remote mirroring storage system, or because it supports remote copy-on-write snapshots.

When a second host can read and write the snapshot image, the second host can start a second Oracle instance. Such a second instance can be used to DSS/reporting purposes. This second instance can also be useful for a limited form of disaster recovery.

DSS/Reporting applications often updates the database. Thus it is important that the second host can write in the snapshot area. For this usage, the snapshot area essentially is a new database started with a copy of the original database. The new database can be created via a cold backup, hot backup, or PITI of the production database. This new database is usually discarded after it has served its purposes. Section 6, 7 and 8 describe how to create a second instance using cold backup, hot backup, and PITI images.

Disaster recovery using remote snapshots is in some sense similar to a second instance on a second host, except that this second instance is never discarded. This second instance becomes the new production database. Section 9 describes how to fail-over and fall back under this usage.

6. Create Second Instance Using Cold Backup

A second oracle database instance can be created on second host by mounting the snapshots from a cold backup. At this point the second instance is totally independent from any other database and can be used for multiple purposes including DSS/reporting.

Use the following steps to create second instance on second host:

- Use Steps described in usage 1 to take local cold backup using storage snapshots
- Let a second host gain access to the snapshot image, possibly by mounting the snapshot.
- Make the snapshot area writeable to the second host
- use initSID.ora from local host and edit as required. This will be used as the new initSID.ora file for the second instance on second host

- startup mount using new initSID.ora file
- if data files reside on different path, rename data files and redo logs as required.
- open database
- use database

7. Create Second Instance Using Hot Backup

Similar to section 6, a second oracle database instance can be created on second host by mounting the snapshots from a hot backup. The steps to accomplish are somewhat different because it involves recovering the database. Consult standard Oracle documentation for information on recovering oracle databases.

Use the next steps to create second instance on second host:

- Use Steps described in usage 2 to take a hot backup using storage snapshots.
- Generate a create controlfile script by issuing ALTER DATABASE BACKUP CONTROLFILE TO TRACE right after ending hot backup mode on production database
- Let a second host gain access to the snapshot image, possibly by mounting the snapshot.
- Make the snapshot area writeable to the second host

- use initSID.ora from local host and make at least the following changes: (1) change control-files parameter so that it will point to a new location accessible by the second host.
- If the second host accesses the datafiles in the snapshot area with a path different from the primary host, edit the create control file creation script so that it points to the right data files.
- create control file with resetlogs option using previous script
- recover database. You must recover enough redo so the database has cleared the hot backup fuzziness. Otherwise the Oracle database won't allow a user to open the database with resetlogs.
- open database with resetlogs option

After one is done with DSS/reporting, one should discard all the changes done by the second instance, including all the archive logs generated by the second instance.

8. Create second Instance on second host by using Point in-time image (PITI)

Point-In-Time Image (PITI) can be accomplished on a second host by starting a second database instance on the remote host. The advantage of a PITI on a second host is that a second database can be started independently of the primary database instance. The second database on the second host can be used for backup purposes and even for DSS reporting and batch processing. One disadvantage of using this method is that that one can not apply archive logs to PITI: PITI diverges from the primary database once it is open.

PITI can be implemented by mounting the snapshot data on a remote host up to the last snapshot taken from a primary local system. Depending on the replication method, the snapshots taken on the local system can be mounted or accessed on the second host.

For PITI to work the snapshot must be ***atomic***. As mentioned before, consult the vendor for more information on atomicity. Furthermore, all data files, all online log files (commonly known as redo logs), and one copy of the control files need to be in the snapshot area.
Use the following steps to create a second instance using PITI:

- Make sure local database is operational
- Split the remote mirror (if using the "snapshot of mirrors" method) or take a new storage snapshot set (if using "copy-on-write method). The snapshot must be atomic.
- Resume operations at primary site
- Use the local snapshot on remote host to startup a second instance. This can be accomplished by manually pushing the snapshots to remote host or any other method the vendors use.
- Mount the image on remote host.
- If all database files reside on the same path as in primary database
    - then just open the database instance. The effect would be just if the database had a database failure and thread recovery was required.
    - No further steps are necessary
- Otherwise, if the data files **are not** in the same path
    - either rename all data files and redolog files by using the ***alter database rename file 'oldfilename' to 'newfilename';*** command
    - or create new control file with ***noresetlogs*** option.
    - open database by issuing the ***Alter database open*** command
- Use new remote database

One challenge common to usage after opening a second instance with this method is to remove the changed blocks on the second host when a re-synchronization is required. When Re-synchronization or refresh from the primary database to the remote host is required, a mechanism to remove the changed blocks on the remote host has to exist. This can be accomplished by taking a snapshot, on the second host, right after the second database is open, then a storage rollback can be applied before a refresh takes place. Some vendor's product already provide this capability automatically. Consult the vendor for specific product features

9. Disaster Recovery

Disaster recovery refers to database recovery from physical disasters such as those caused by power outage, fire, flood, earthquake, etc. It is important to develop a test plan for recovering the primary database services in the event of system failure due to a disaster incident. The more time and effort is invested in creating and testing the disaster recovery plan, the better prepared the organization would be should disaster strike.

Database disaster recovery typically requires that the primary production database be replicated at a remote site. There are various ways and products of implementing High Availability (HA) in the market, including Oracle Fail Safe, Oracle Parallel Server, UNIX Clustering, NT Clustering, Oracle HA Server and others.

Storage Snapshots can also be used for implementing a **very limited** HA solution. To achieve the best results, the Oracle primary database should have reasonable frequent storage snapshots taken. Furthermore, it is recommended that the snapshot images reside on a remote facility. The recovery process using snapshot images would only recover up to the last snapshot image taken of the system prior to the time of failure. This solution is only as good as the frequency at which storage snapshots are taken. With this approach, there will always be transaction loss. One can use the steps described in sections 6,7 & 8 to implement a disaster recovery solution.

Another method, for a limited disaster recovery solution, would be by using **periodic** remote cold and hot backup snapshots. This can be implemented just as described in sections 6 and 7, except that the second instance is not discarded,

Fail-over Process
The process to switch from a primary database to a secondary database due to system failure is called ***fail-over***.

One approach for implementing ***fail-over*** using storage snapshots is to use ***PITI***, as explained in section 8. As mentioned earlier, this new database instance is used as the primary database and thus any changes to it **will not** be discarded.

After opening the new database a full backup has to performed and all previous **ARCHIVELOGS** need to be discarded. Old Archive logs are not usable after a resetlogs operation is performed when opening a database.

This approach assumes that the local snapshots are mounted with read-write option on a second host.

For ***fail-over*** method use the following steps:

- Use the local snapshot (taken from the local host)on remote host to startup a second instance. Use steps in section 8.
- Open database using resetlogs option
- Take a full database backup (hot or cold)
- Remove old archive logs on the second host
- Resume Operations with new production database

Now the secondary host serves as the primary database production environment.

Fall Back Process
Once the damaged system has been repaired, there might be the need to switch back to a prior configuration. This process is called a ***fall back*** operation, where the recently repaired production environment becomes once again a production environment.

One of the biggest challenges for a ***fall back*** operation is to re-sync both environments again without affecting performance on the current production environment. This is very critical for systems where the mode of operation is 7/24.

One solution to the problem would be by using storage snapshots incrementally to re-sync both systems. When the actual ***fall back*** action (switch over from current primary system to new primary system) takes place, the down time is minimal.

Explicit step by step instructions on how to implement ***fall back*** procedures are not described in this paper. It is beyond the scope of this paper to provide step by step information on how to ***fall back*** a system after a ***fail-over*** has taken place.

References

[1] Guidelines for Using Remote Mirroring Storage Systems for Oracle Database, J. Bill Lee, Oracle Corporation.

[2] The High Availability Database Server Cookbook, Bob Thome, Oracle Corporation.

[3] ORACLE Backup & Recovery Handbook, Rama Velpuri, Oracle Press

Acknowledgments

Kathie Mercier, Oracle Corporation, Oracle Consulting Services