How to Implement a *"snapshot standby"* Using Oracle Data Guard 10*g* Release 2

"We get double-duty from our Data Guard physical standby databases by using them as test instances. This increases return on investment in our disaster recovery systems and dramatically improves the quality of test results, reducing business risk when implementing changes in our production environment"

Sreekanth Chintala[1]
Database Engineering, Dell Inc.

Technologies Used

- Oracle Database 10g Release 2
- Oracle RAC
- Oracle Data Guard
- Oracle Flashback Database
  - Guaranteed Restore Points
- Oracle Recovery Manager (RMAN)

EXECUTIVE SUMMARY

Dell has utilized Oracle Data Guard 10*g* Release 2 and related Oracle Database capabilities to utilize their physical standby databases for testing and other purposes that require read-write access to the standby database. This has increased Dell's return on investment in Disaster Recovery (DR) systems. As important, the increased accuracy of test results has dramatically decreased business risk when changes must be made to their production environment. Dell has implemented via a manual process the near equivalent of the new Data Guard Snapshot Standby feature delivered in Oracle Database 11*g*. Oracle Database 10*g* users can follow the steps provided in this paper to realize the benefit of a snapshot standby database without having to wait for Oracle Database 11*g* to be production in their shops.

OVERVIEW

Every mission critical system should have a DR plan in place to provide business continuity. Though disaster recovery systems require significant investment in storage, servers, network bandwidth, physical space etc., the cost to the business of extended downtime or data loss is even greater. Finding a way to utilize these mostly unused DR resources while they are in standby role can significantly increase your return on investment (ROI) in standby systems.

It is not unusual for non-production environments, such as development, functional/performance/system integration testing, to not be on par with the production environment in terms of cpu power, storage and memory. Access to production data is also required in order to achieve meaningful test results – thus additional effort and expense is incurred populating test databases. Testing

---

[1] ***About the Author*** *Sreekanth Chintala works for Dell Inc., and is an OCP Certified DBA who has been using Oracle technologies for 10 years and has over 15 years of I/T experience. Sreekanth specializes in Oracle High Availability, Disaster Recovery solutions and developing emerging Oracle technologies for Dell IT. Sreekanth received his Master's (M.S) from West Virginia University, Master of Technology (M.Tech) from REC Calicut (India) and a Bachelor's degree from JNT University, Hyderabad (India). Sreekanth may be reached at sreekanthchintala@gmail.com*

performed on an inferior infrastructure and using stale data or a subset of the production database is unlikely to provide realistic test results.

Oracle Database 11*g* includes a new Data Guard feature "Snapshot Standby" that enables users to significantly boost their ROI in DR systems by easily utilizing them for test purposes without compromising data protection. But many users can't afford to wait until they upgrade to Oracle 11*g*. The good news is that users can create their own snapshot standby with manual procedures using Oracle Database 10*g* Release 2.

Creating an Oracle Data Guard 10*g* Release 2 snapshot standby opens up new avenues for DBAs. Existing DR infrastructure can be used for testing, performance turning or a benchmarking environment while continuing to provide protection from disasters. Some of the benefits of using a snapshot standby for testing are:

- Eliminate guesswork. Performance test results are realistic and reliable
- Know the impact. Realistic system resource consumption on production data during performance testing
- Query tuning is no longer dependent on best execution plan from a subset of old data
- Consolidate non-production systems by reducing test environments
- Save time and effort in populating test environments frequently
- Change assurance. Introduce changes with confidence

This paper provides step-by-step instructions to open a Data Guard 10*g* Release 2 physical standby database for read-write operations, and then convert it back to its original state as a synchronized physical standby database. Oracle Flashback Database and Guaranteed Restore Points are used in conjunction with Data Guard to resynchronize the standby database once read-write operations are complete. The standby database in this example is configured with Oracle RAC, ASM and the Data Guard Broker, though the same procedure applies to a non-RAC configuration.

FLASHBACK DATABASE

Flashback Database [1] is used to rewind a database back in time using its own logs called Flashback Database Logs. When you use Flashback Database, Oracle Database uses past block images to back out changes to the database. During normal database operation, Oracle Database occasionally logs these block images in flashback logs. This is done using a new background process called Recovery Writer (RVWR). These block images can later be used to reconstruct the data file contents as of any point in time at which logs were captured. In addition to the new background process, RVWR, a new buffer in the SGA, called Flashback Buffer is used support flashback feature. Flashback log files are written to the Flash Recovery Area [2].

Enabling Flashback Database

The following instructions on enabling flashback database apply to a primary or a standby database. Steps related to ASM and Data Guard Broker can be ignored if you are not using them. Note: If you are already using a Flash Recovery Area (FRA), skip to step 7, below.

1. Identify the lun(s) that can be used in creating a diskgroup that serves as FRA. If you are using an existing diskgroup that is already created, skip to step 5.

2. Create and mount a diskgroup in ASM that serves as the FRA. The Diskgroup can be created from any node of the cluster.

   This example uses two luns: /u02/oradata/asm/lun4 and /u02/oradata/asm/lun5

```
$ export ORACLE_SID=+ASM1
$ sqlplus "/ as sysdba"

SQL> create diskgroup FRA '/u02/oradata/asm/lun4',
     '/u02/oradata/asm/lun5' ;
SQL> alter diskgroup FRA mount ;
```

3. Check if the FRA diskgroup is mounted and note down the space available in this diskgroup.

```
$ export ORACLE_SID=+ASM1
$ sqlplus "/ as sysdba"
```

```
SQL> show parameter diskgroup

NAME             TYPE        VALUE
---------------- ----------- ------------------------------
asm_diskgroups   string      DATA, FRA, ARCH
SQL>
```

```
Set linesize 132
col path for a22
col name for a15
col DG_NAME for a11
col DISK_NAME for a11
select b.name DG_NAME , a.name DISK_NAME, a.path, a.total_mb,
a.free_mb from v$asm_disk a, v$asm_diskgroup b
where a.GROUP_NUMBER = b.GROUP_NUMBER
order by 1;

DG_NAME     DISK_NAME   PATH                    TOTAL_MB    FREE_MB
----------- ----------- ---------------------- ---------- ----------
DATA        DATA_0000 /u02/oradata/asm/lun2      133120     107842
DATA        DATA_0001 /u02/oradata/asm/lun1      186368     160436
ARCH        ARCH_0000 /u02/oradata/asm/lun3      186368     186317
FRA         FRA_0000  /u02/oradata/asm/lun4      204800     173556
FRA         FRA_0001  /u02/oradata/asm/lun5      204800     173556
```

4. Make sure the new diskgroup is added to ASM initialization parameter file on ALL RAC nodes so that FRA disk group is mounted whenever an ASM instance is bounced.

This example shows that you are adding 'FRA' diskgroup to "disk_groups" parameter that already has 'DATA' and 'ARCH' assigned to it.

disk_groups = 'DATA', 'ARCH', 'FRA'

5. Make sure ORACLE_SID is pointing to the database instance. Enable archive log mode if the database is not in archive log mode. It is a requirement for the database to be in archive log mode. In this example, the archive log destination is the ARCH diskgroup in ASM.

```
SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            +ARCH
Oldest online log sequence     442
Next log sequence to archive   443
Current log sequence           443
```

6. Set FRA related initialization parameters. The following initialization parameters need to be set to enable FRA.

DB_RECOVERY_FILE_DEST specifies the default location for the FRA. The FRA could contain multiplexed copies of current control files and online redo logs, as well as archived redo logs, flashback logs and RMAN backups. DB_RECOVERY_FILE_DEST_SIZE (in bytes) specifies the hard limit on the total space to be used by target database recovery files created in the FRA.

DB_FLASHBACK_RETENTION_TARGET (default 1440) specifies the upper limit (in minutes) on how far back in time the database may be flashed back. How far back one can flashback a database depends on how much flashback data Oracle has kept in the FRA. It is important to note that the DB_FLASHBACK_RETENTION_TARGET is a target, not an absolute guarantee. If the FRA is not large enough to hold required files such as archived redo logs and other backups, then flashback logs may be deleted to make room for these required files. If you have a standby database, then set DB_FLASHBACK_RETENTION_TARGET to be the same for both primary and standby databases. V$FLASHBACK_DATABASE_LOG can be queried to determine how far you can flashback.

```
$  sqlplus "/ as sysdba"

SQL> alter system set db_recovery_file_dest_size=200G
     scope=both sid='*'

SQL> alter system set db_recovery_file_dest='+FRA'
     scope=both sid='*' ;
```

```
SQL> alter system set db_flashback_retention_target=2880
     scope=both sid='*' ;
```

```
SQL> show parameter db_recovery

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------
db_recovery_file_dest                string      +FRA
db_recovery_file_dest_size           big integer 200G
```

7. Stop Data Guard Broker if it is running. If Data Guard Broker is configured, the Broker brings up the primary or standby database when it is mounted. You don't want the Data Guard Broker to bring up the Standby database while enabling Flashback database.

```
SQL> alter system set dg_broker_start='FALSE' scope=both
     sid='*';
```

8. Shutdown and startup the database in MOUNT EXCLUSIVE mode to turn on Flashback database.

```
$ srvctl stop database -d <db_name>
SQL> STARTUP MOUNT EXCLUSIVE;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN ;
```

9. Determine if Flashback database is enabled

```
SQL> select flashback_on from  v$database;
FLASHBACK_ON
------------
YES
```

10. Start Data Guard Broker if it was running earlier.

SQL> alter system set dg_broker_start= 'TRUE ' scope=both sid= '*' ;

11. Flashback database log activity can be monitored using the following query:

```
SQL> select begin_time, flashback_data, db_data, redo_data,
    ESTIMATED_FLASHBACK_SIZE from v$flashback_database_stat;
```

12. Flashback database retention target details can be monitored using the following query:

```
SQL> select * from   v$flashback_database_log ;
```

RESTORE POINTS

Restore Point is an Oracle Database 10*g* feature that provides the ability to associate a user-defined name, for example, TEST_NEW_FEATURE, with a point-in-time SCN number to establish a restore point that can be used with Flashback database or Flashback table during recovery operations. A restore point can be specified such that it guarantees the database can be recovered to a particular point-in-time and eliminates the need to manually record an SCN or timestamp to use with Flashback database and Flashback table operations. Beginning with Oracle Database 10*g* Release 2 a "Guaranteed Restore Point" ensures that the database can be flashed back to that particular point-in-time regardless of the flashback retention target.

COST BENEFIT ANALYSIS

Every useful feature comes with certain cost and risk associated with it. Enabling Flashback database requires additional space to hold flashback logs. Enough space will also need to be allocated to hold archive logs generated by the primary database while the standby database is open read-write. In Oracle Database 10*g* Release 2, while the physical standby database is activated redo data from the primary database cannot be shipped or applied to the standby database. This requires additional arrangements to be made to continue to provide complete data protection (e.g. creation of an archive log repository at the standby location). However, once the activated database is converted back to a physical standby database, Data Guard can automatically ship and apply all the needed archivelogs from the point of divergence.

Dell has determined that the benefits of using the snapshot standby procedures documented in this paper for test purposes far outweigh the cost and risk involved in opening a physical standby database for read/write operations.

ADVANCE PREPARATIONS

Before opening the physical standby, planning is required to understand how long the standby will be in read/write mode, the storage required to hold the archive logs generated by the primary database during this time, and the time required to resynchronize the standby database once test operations are complete. It is recommended to keep archivelogs on the disk to keep resynchronization time to a minimum. Metrics provided by v$archived_log view can be used to calculate how long you can keep archivelogs on disk. It is wise to pick a time when there is reduced activity on the primary.

DATA GUARD ENVIRONMENT

The configuration used in this example is as follows:

Primary database	Physical Standby database:
2-node 10*g*R2 (10.2.0.2) RAC ASM for all data storage	2-node 10*g*R2 (10.2.0.2) RAC ASM for all data storage
Linux	Linux
Data Diskgroup: DATA	Data Diskgroup: DATA
Archivelog diskgroup: ARCH	Archivelog diskgroup: ARCH
FlashRecoveryArea diskgroup: FRA	FlashRecoveryArea diskgroup: FRA
Data Guard Broker enabled	Data Guard Broker enabled

Note: The above example assigns archive logs to their own ASM disk group. Customary Oracle best practices recommend that archive logs are also placed in the Flash Recovery Area such that they are managed automatically by Oracle. Dell's current preference is to manage them manually.

Figure 1 shows a Data Guard environment that consists of a 2-node Primary, 2-node physical standby and Data Guard Broker managing the configuration.

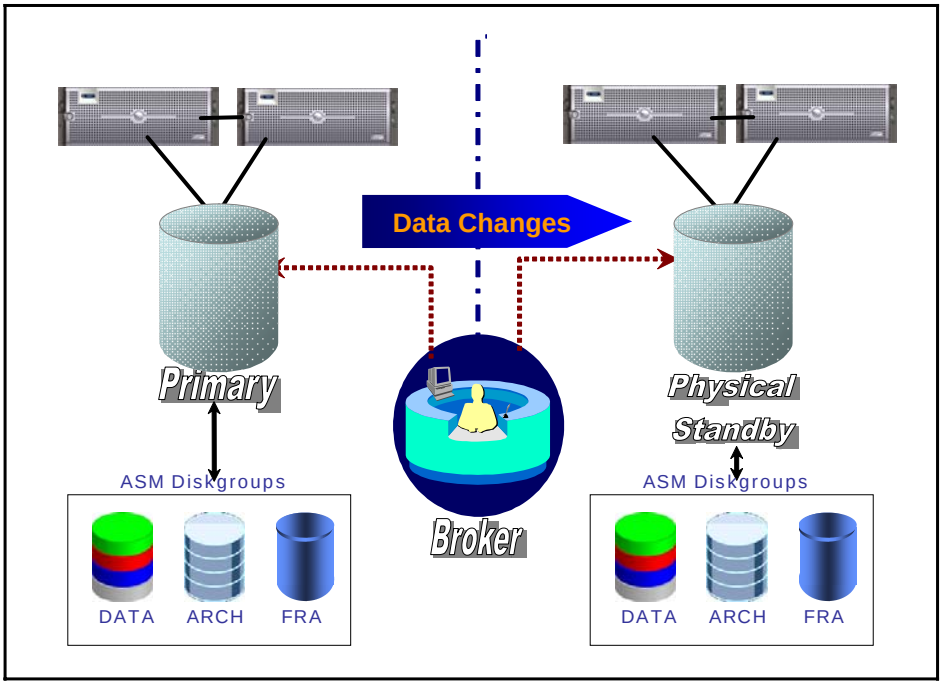

Figure 1: Data Guard environment

REQUIREMENTS

The following requirements need to be met in order to create a snapshot standby.

- The Data Guard environment (a primary and a physical standby) needs to be Oracle 10.2.0.1 or higher
- Primary and Standby databases need to be in archivelog mode (the default requirement for Data Guard)
- Force logging should be set to TRUE to avoid no-logging operations
- Flash Recovery Area (FRA) is required on the standby database to implement flashback database. Note that Oracle best practices recommend implementing an FRA and Flashback database on the primary as well, though this is not required for the snapshot database operation described in this paper.
- Primary and standby are in sync at the time of the test or the gap between primary and physical standby is nominal

Figure 2 shows a high level picture of steps involved in creating a Snapshot Standby database. These steps can be repeated as many times as necessary.

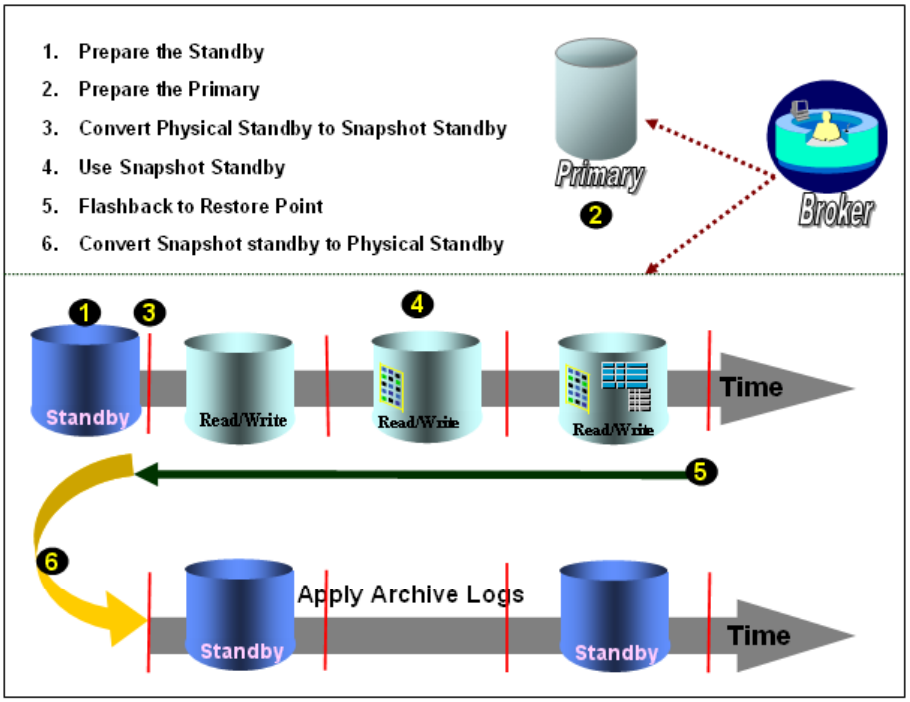

Figure 2: Snapshot Standby (open physical standby read/write)

CREATING AND USING A SNAPSHOT STANDBY

Now we will look in to detailed steps on how to open physical standby database for read/write operations and revert back to a physical standby database.

Prepare the physical standby database to be activated

1. Make sure the standby and primary are in sync.
2. Stop Data Guard Broker on the standby database. If Data Guard Broker is configured, the Broker would restart the apply process when it is mounted. Setting the Data Guard Broker to FALSE prevents this from happening. You don't want the Data Guard Broker to attempt to start applying redo during this operation.

```
SQL> alter system set dg_broker_start=FALSE scope=both
     sid='*' ;
```

3. Shutdown all non-applying standby instances on the Oracle RAC. Even though there may be multiple instances on a Oracle RAC standby database, only one of the instances will be receiving and applying the archived logs.
4. If there are no other cascading remote log destinations defined, ignore this step. If there are cascading standby destinations defined, defer these log archive destinations. For example, if log_archive_dest_3 is defined to send archive logs to another physical standby then defer that shipping by;

```
SQL> alter system set log_archive_dest_state_3=DEFER
     scope=both sid='*' ;
```

5. Note down the time and SCN number (as a safety measure), cancel the recovery on the applying instance and create a "Guaranteed Restore Point"

```
SQL> Select DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER()
     from dual;

SQL> alter database recover managed standby database
     cancel;

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

Prepare the primary database

6. Archive the current log file. On the primary database, switch logs so the SCN of the restore point will be archived on the physical standby database. When using the standby redo logs, this step is ESSENTIAL to ensure the database can be properly flashed back to the restore point.

```
SQL> alter system archive log current ;
SQL> alter system switch logfile ;
SQL> alter system switch logfile ;
```

7. While the Physical Standby is open for read/write operations, redo shipping cannot continue. You must defer the redo transport destination for the physical standby database. For example, if log_archive_dest_2 is defined to send archive logs to the physical standby then defer that shipping by,;

```
SQL> alter system set log_archive_dest_state_2=DEFER
     scope=both sid='*' ;
```

8. The archivelogs should be kept on-disk either on primary or on standby. If the archivelogs need to be backed up and deleted, be advised that these archivelogs need to be restored after the physical standby is back in recovery mode or else plan on using an incremental backup to synchronize the physical standby database after it is reverted back to its original role.

Activate the Physical Standby

9. Activate the physical standby and modify the protection mode if necessary. If the standby database is either "Max Availability" or "Max Protection" mode, downgrade the protection mode to "Maximum Performance" mode. Note: If the Data Guard configuration is running in Maximum Protection and this is the only standby you cannot use this procedure as converting the standby will cause the Primary database to crash.

```
SQL> alter database activate standby database;
SQL> startup mount force;
SQL> alter database set standby database to maximize
performance;
SQL> alter database open;
```

Use the activated database

Once the standby database has been activated, it is a full-blown production system. You may run reports or perform testing or test new code or create objects. There are no limitations on what you can or cannot do on the activated database, other than the few restrictions that relate to Flashback Database [3]. Any results stored in the activated database will be lost when the database is flashed back to before activation time. Results that should be saved must be copied or exported out of the activated database before flashing it back.

Revert the Snapshot database to Physical Standby

10. After the testing is completed, you need to resynchronize the activated database with the primary database. Issue the following statements on the activated database to quickly flashback to the Guaranteed Restore Point. Data

Guard will resynchronize it with the primary database.

```
SQL> STARTUP MOUNT FORCE;
SQL> FLASHBACK DATABASE TO RESTORE POINT TEST_NEW_FEATURE;
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH
     SID='*';
SQL> STARTUP MOUNT FORCE;
```

11. If the standby was open for a long time, and the archivelogs are not available on disk either on the primary or on the standby but they were backed up and deleted, you need to restore the archivelogs to the standby or else use an incremental backup to synchronize the Physical standby database.

12. If there were any cascading databases that were feeding off of this physical standby, re-enable those destinations.

Re-enable log shipping on Primary

13. Enable the archive log destinations that were DEFERed. For example, if log_archive_dest_2 is defined to send archive logs to the physical standby then enable that shipping by,;

```
SQL> alter system set log_archive_dest_state_2=ENABLE
     scope=both sid='*' ;
```

OPTIONAL: UTILIZE AN ARCHIVE LOG REPOSITORY ON THE STANDBY SERVER

The most significant limitation of the manual Data Guard 10*g* process for creating a snapshot standby is the fact that redo transport is suspended while the standby database is open read-write. The procedures above require the primary redo data generated during this period to be archived on-disk at the primary database in order for Data Guard to automatically resynchronize the standby database with the primary database once the flashback operation is complete. This obviously subjects data to potential loss if a failure of the primary system occurs before resynchronization is complete.

To protect data while in this state, and to accelerate the resynchronization process, an Archive Log Repository can be created on the standby server. This will allow the primary database to continue shipping current redo to the standby location, where it will be safely archived on disk, and available locally to the standby database when the resynchronizations process is begun. For more details on creating and using an Archive Log Repository, please refer to Oracle documentation in chapter 5 of the Oracle Data Guard Concepts and Administration documentation for Oracle Database 10g Release 2 [4].

LOOKING AHEAD TO ORACLE DATABASE 11*g*:

The new Oracle Database 11*g* Feature, *Data Guard 11g Snapshot Standby*, automates the manual steps described in this paper via a single Data Guard Broker command or a single mouse-click when using Enterprise Manager Grid Control. In addition to automation, Data Guard 11g Snapshot Standby provides continuous data protection; even while the snapshot standby database is open read-write.

Additional benefit is realized by using Snapshot Standby in combination with a second new Oracle Database 11*g* feature called "Database Replay". Database Replay allows you to capture the workload on a production system that can be replayed many times on a test system. Database Replay provides the ability to run production workloads that may include online and batch workloads in the non-production environment. Database Replay enables you to assess the impact of a change by replaying the captured production workload on the test system.

Replaying production workload on a test system that doesn't have production data wouldn't yield realistic results. Snapshot Standby compliments Database Replay as it offers production data and production infrastructure for testing. DBAs now have a full-blown production infrastructure at their disposal to test their changes without the overhead of having to duplicate an entire application infrastructure.

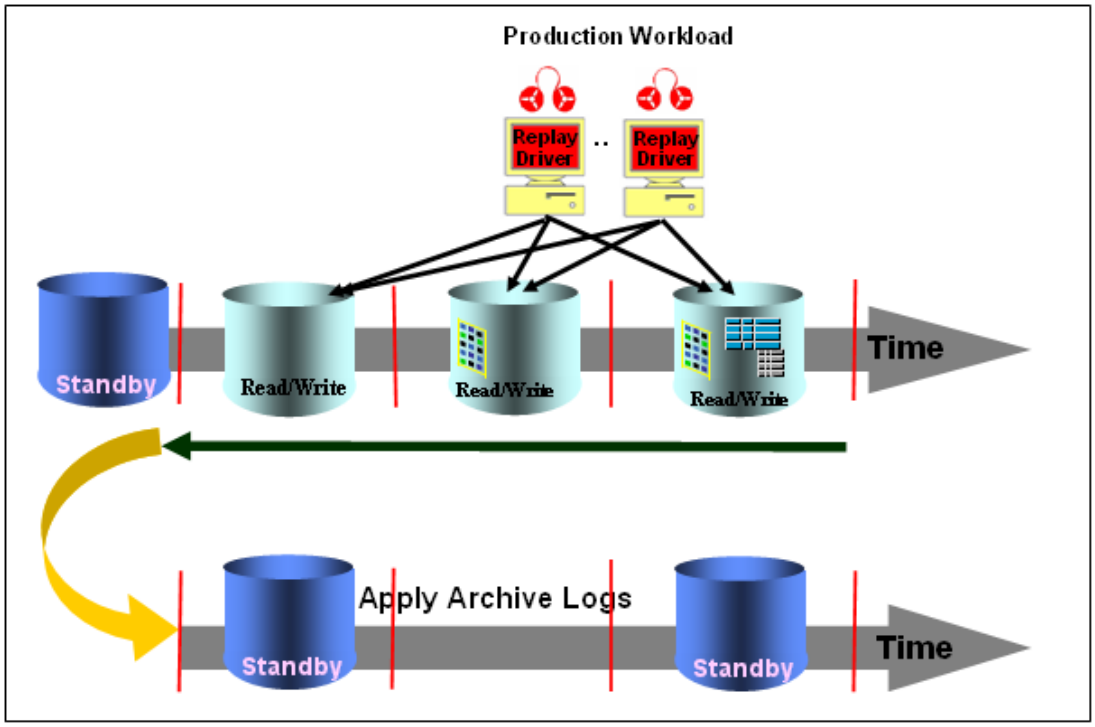

Figure 3: Snapshot Standby and Database Replay (11g)

SUMMARY

Having read/write access to production data opens up a variety of avenues for DBAs and database developers. Without realistic test results, estimating the impact of introducing a change on a production system is often a best guess. Testing performed on a non-production system, using a stale copy of the production database or a limited subset of production data may not give true impact of the changes to be introduced. A Data Guard snapshot standby provides as a test environment that enables DBAs to assess the impact of the changes before implementing them on production.

Don't wait for Oracle Database 11*g* to be production in your shop. Use the steps outlined in this paper to create your own snapshot standby using Data Guard 10*g* Release 2 today. Achieve maximum ROI on your DR infrastructure by utilizing your standby systems as a ready-made test infrastructure having real production data and performance characteristics. Most importantly, the increased accuracy of test results will dramatically decrease business risk when changes must be made to your production environment.

REFERENCES

1. Flashback Database
   http://www.oracle.com/technology/deploy/availability/htdocs/Flashback_Overview.htm
2. Flash Recovery Area
   http://www.databasejournal.com/features/oracle/article.php/3446681
3. Flashback Database – additional information
   http://download.oracle.com/docs/cd/B19306_01/backup.102/b14192/rpfbdb003.htm#sthref508
4. Data Guard Archive Log Repository – see section 5.2.1, Destination Types
   http://download.oracle.com/docs/cd/B19306_01/server.102/b14239/log_transport.htm

September 2007
Author: Sreekanth Chintala – Dell Inc.

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com