Using a 10g R2 Oracle Physical Standby Database For Read/Write Testing and Reporting

Alejandro Vargas
Oracle Support Israel
Principal Support Consultant

Introduction

On 11g we can open a Physical Standby database on read-only mode, while the Standby continues to be updated. This is a major break through on the High Availability field.

On 10g R2 we can take advantage of 10g Flashback Database together with a Physical Standby to be able to open the Standby Database on read-write mode, perform testing / reporting activities and, once these activities are completed we can reinstate the standby and synchronize it with the primary database.

This document is a step by step implementation test of this technology and is based on
The Oracle Data Guard Concepts and Administration, 10g Release 2 guide, Chapter 12, Data Guard Scenarios.

Description of the Test

This test is implemented using 2 Linux servers, and Oracle 10g R2 EE.

- Primary database ‘dgedb’ is located on server rac1.
- Physical standby database ‘dgfdb’ is located on server rac2

We will activate and open the standby read – write, and after simulating some activity, we will reinstate it as physical standby.

First Part: Prepare the environment and activate read/write the standby database.

1. Check that primary and standby are synchronized
2. Check that standby is applying redo logs
3. Enable flashback database on physical standby
4. Stop redo apply to the standby

5. Create a restore point on the standby.
6. Archive log current on primary database
7. Defer redo log shipping to the standby.
8. Activate the physical standby database.
9. Startup mount force the standby.
10. Set the standby protection mode to maximum performance.
11. Open the activated standby database
12. Use the activated standby for reporting/testing

Second part: Resynchronize the Physical Standby Database with the Primary Database

13. Revert activated database to physical standby role
14. Startup mount force
15. Flashback activated database to restore point
16. Convert activated database to physical standby
17. Startup mount force
18. Put the standby database on recover mode.
19. Enable archive log shipping to the standby
20. Check that archive logs are applied to the standby

End of the procedure

Implementation Step by Step

1. Check that primary and standby are synchronized

On this step we check that primary and standby are synchronized, by checking the current log sequence, using the archive log list command.

PRIMARY

```
ORACLE_SID=dgedb
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1

SQL> archive log list

     Database log mode Archive Mode
     Automatic archival Enabled
     Archive destination /u01/app/oracle/oradata/dgedb/archive/
     Oldest online log sequence    27
     Next log sequence to archive  29
     Current log sequence          29

SQL> select database_role, open_mode from v$database;

     DATABASE_ROLE    OPEN_MODE
     ---------------- ----------
     PRIMARY          READ WRITE
```

PHYSICAL STANDBY

```
ORACLE_SID=dgfdb
ORACLE_BASE=/u01/app/oracle
```

```
ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1


SQL> archive log list;

     Database log mode Archive Mode
     Automatic archival Enabled
     Archive destination /u01/app/oracle/oradata/dgfdb/archive/
     Oldest online log sequence    29
     Next log sequence to archive  0
     Current log sequence          29

SQL> select database_role, open_mode from v$database;

     DATABASE_ROLE OPEN_MODE
     ---------------- ----------
     PHYSICAL STANDBY MOUNTED
```

2. Check that standby is applying redo logs

On this step we will make switch logfile on the primary and will check how the sequence is advanced also on the physical standby

PRIMARY

```
SQL> alter system switch logfile;
```

```
     System altered.

SQL> alter system switch logfile;

     System altered.

SQL> archive log list

     Database log mode Archive Mode
     Automatic archival Enabled
     Archive destination /u01/app/oracle/oradata/dgedb/archive/
     Oldest online log sequence    31
     Next log sequence to archive  33
     Current log sequence          33
```

PHYSICAL STANDBY

```
SQL> archive log list

     Database log mode Archive Mode
     Automatic archival Enabled
     Archive destination /u01/app/oracle/oradata/dgfdb/archive/
     Oldest online log sequence    31
     Next log sequence to archive  0
     Current log sequence          33
```

3. Enable flashback database on physical standby

On this step we will set up a flash recovery area, flashback log will be written to the db_recovery_file_dest, they will enable the possibility to flashback the database to the point in time we will set in order to re-activate the physical standby once the test is finished.

```
SQL> alter system set db_recovery_file_dest_size=2g scope=both;

System altered.

SQL> alter system set db_recovery_file_dest='/u01/app/oracle/flashback'
scope=both;

System altered.

SQL> show parameters db_recovery

NAME TYPE VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest string /u01/app/oracle/flashback
db_recovery_file_dest_size big integer 2G
```

To enable flashback database the database needs to be shutdown cleanly, after that start up mount and enable flashback.

```
SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database flashback on;
```

```
Database altered.
```

4. Stop redo apply to the standby

On the physical standby database we need to stop Redo Apply.

```
SQL> alter database recover managed standby database cancel;

Database altered.
```

5. Create a restore point on the standby

A guaranteed restore point will make possible for us to flashback the database to this point in time, once our read write activities on its activated version will finish.

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

Restore point created.
```

6. Archive log current on primary database

We need to archive the current redo log on the primary database; it will be shipped to the standby server but not applied. This way we will assure that when flashing back and reactivating the standby we will have the archived logs up to the SCN of the restore point.

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

```
System altered.

SQL> archive log list;
Database log mode Archive Mode
Automatic archival Enabled
Archive destination /u01/app/oracle/oradata/dgedb/archive/
Oldest online log sequence    34
Next log sequence to archive  36
Current log sequence          36
```

7. Defer redo log shipping to the standby.

This step is performed on the Primary Database; we defer redo log shipping to the standby while it will be activated read write.

First we check the actual values for the standby archive log destination

```
SQL> show parameters log_archive_dest_state_2

NAME TYPE VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_state_2 string ENABLE

SQL> show parameters log_archive_dest_2

NAME TYPE VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_2 string SERVICE=dgfdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=dgfdb
```

Once we are sure which is the destination to defer we can execute the defer command

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=DEFER;

System altered.

SQL> show parameters log_archive_dest_state_2

NAME TYPE VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_state_2 string DEFER
```

8. Activate the physical standby database.

At this moment the physical standby has stopped redo apply, we have enable flashback database and we have created a restore point. We are ready to activate the database and open it for read/write operations.

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;

Database altered.
```

9. Startup mount force the standby

```
SQL> STARTUP MOUNT FORCE;
ORACLE instance started.
```

```
Total System Global Area 218103808 bytes
Fixed Size                  1260984   bytes
Variable Size             134218312 bytes
Database Buffers           79691776  bytes
Redo Buffers               2932736   bytes
Database mounted.
```

10. Set the standby protection mode to maximum performance

We need to downgrade the protection mode to maximum performance because this mode permits to work without a standby database protecting the activated standby once opened.

When later we will flash back to the restore point, the physical standby automatically will get the protection mode defined on the primary.

```
SQL> alter database set standby database to maximize performance;

Database altered.
```

11. Open the activated standby database

```
SQL> ALTER DATABASE OPEN;

Database altered.
```

12. Use the activated standby for reporting/testing

Now we have the standby database activated. It is a block by block 100% copy of our production environment we can use it to run reports, to do applications testing, i.e. implementing a new version of our applications and testing it, perform critical changes to the structure, if required on production and test the impact, etc.

NOTE: While the standby is open read write do not provide protection to the Primary. You may have a second standby for this purpose.

In this test I'm executing some read / write operations to mimic some reporting activity

```
SQL> connect avargas/oracle
Connected.

SQL> create table myusers as select * from dba_users;

Table created.

SQL> insert into myusers select * from myusers;

22 rows created.

SQL> /

44 rows created.

SQL> /

88 rows created.

SQL> /

176 rows created.
```

```
SQL> /

352 rows created.

SQL> commit;

Commit complete.

SQL> select username from myusers where username like 'AVAR%';

USERNAME
------------------------------
AVARGAS
…
…
…
AVARGAS
AVARGAS

32 rows selected.
```

13. Revert activated database to physical standby role

On this step we will revert the activated database to its standby role, note that all information stored during the time the database was open read/write will be lost. You may want to make a backup if you need this information for later use.

3 steps are required to revert to the standby role:

- Startup mount force
- Flashback database
- Convert to physical standby

14. Startup mount force

Note the log sequences we have on the standby, we are as a matter of fact on a new incarnation

```
SQL> startup mount force;
ORACLE instance started.

Total System Global Area 218103808 bytes
Fixed Size                  1260984 bytes
Variable Size             142606920 bytes
Database Buffers           71303168 bytes
Redo Buffers                2932736 bytes

Database mounted.

SQL> archive log list

Database log mode Archive Mode
Automatic archival Enabled
Archive destination /u01/app/oracle/oradata/dgfdb/archive/
Oldest online log sequence    1
Next log sequence to archive  2
Current log sequence          2
```

15. Flashback activated database to restore point

On this step we are using the restore point `Before_App_Test` we created on step 5. We may have used also the SCN or a point in time.

```
SQL> flashback database to restore point before_app_test;

flashback complete.
```

16. Convert activated database to physical standby

SQL> alter database convert to physical standby;

Database altered.

17. Startup mount force

After having converted the database to the physical standby role we need to restart the database, notice that still at this point the sequences are after the restore point

```
SQL> STARTUP MOUNT FORCE;
ORACLE instance started.

Total System Global Area      218103808 bytes
Fixed Size                      1260984 bytes
Variable Size                 150995528 bytes
Database Buffers               62914560 bytes
```

```
Redo Buffers                    2932736 bytes
Database mounted.

SQL> archive log list;

Database log mode Archive Mode
Automatic archival Enabled
Archive destination /u01/app/oracle/oradata/dgfdb/archive/
Oldest online log sequence    1
Next log sequence to archive  2
Current log sequence          2
```

18. Put the standby database on recover mode.

When enabling recovery the standby database will start to apply archived logs from the primary that will take a minute until redo shipping is enabled on the primary.

```
SQL> alter database recover managed standby database disconnect;

Database altered.
```

On the alert log of the standby database we can see that the online redo logs are cleared from the sequences they were working on while opened read/write

```
Thu Dec 27 17:30:17 2007
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT
Thu Dec 27 17:30:17 2007
Attempt to start background Managed Standby Recovery process (dgfdb)
MRP0 started with pid=28, OS id=27288
```

```
Thu Dec 27 17:30:18 2007
MRP0: Background Managed Standby Recovery process started (dgfdb)
Managed Standby Recovery not using Real Time Apply
parallel recovery started with 2 processes
Clearing online redo logfile 1 /u01/app/oracle/oradata/dgfdb/redo01.log
Clearing online log 1 of thread 1 sequence number 2
Thu Dec 27 17:30:24 2007
Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT
Thu Dec 27 17:30:27 2007
Clearing online redo logfile 1 complete
Clearing online redo logfile 2 /u01/app/oracle/oradata/dgfdb/redo02.log
Clearing online log 2 of thread 1 sequence number 1
Clearing online redo logfile 2 complete
Media Recovery Waiting for thread 1 sequence 35
```

19. Enable archive log shipping to the standby

On the primary database we enable log shipping to the standby, when the stream of changes get to the standby its online logs will be cleared from the sequences stored on them when it was activated and open on read write mode

```
SQL> alter system set log_archive_dest_state_2=enable;

System altered.

SQL> select database_role, open_mode from v$database;

DATABASE_ROLE OPEN_MODE
---------------- ----------
```

```
PRIMARY READ WRITE

SQL> alter system switch logfile;

System altered.

SQL> archive log list;

Database log mode Archive Mode
Automatic archival Enabled
Archive destination /u01/app/oracle/oradata/dgedb/archive/
Oldest online log sequence    35
Next log sequence to archive  37
Current log sequence          37
```

20. Check that archive logs are applied to the standby

On step 19, after enabling redo shipping to the standby we did a switch logfile, now we can check if the standby is in sync with the primary, the current log sequence needs to be 37.

```
SQL> select database_role, open_mode from v$database;

DATABASE_ROLE OPEN_MODE
---------------- ----------
PHYSICAL STANDBY MOUNTED

SQL> archive log list

Database log mode Archive Mode
```

```
Automatic archival Enabled
Archive destination /u01/app/oracle/oradata/dgfdb/archive/
Oldest online log sequence    0
Next log sequence to archive  0
Current log sequence          37
```

On the standby alert.log we can see the Remote File Server (RFS) process executing the apply process up to the last sequence

```
Thu Dec 27 17:34:28 2007
Using STANDBY_ARCHIVE_DEST parameter default value as
/u01/app/oracle/oradata/dgfdb/archive/
Redo Shipping Client Connected as PUBLIC
-- Connected User is Valid
RFS[1]: Assigned to RFS process 1352
RFS[1]: Identified database type as 'physical standby'
Primary database is in MAXIMUM PERFORMANCE mode
Thu Dec 27 17:34:28 2007
RFS LogMiner: Client disabled from further notification
Primary database is in MAXIMUM PERFORMANCE mode
RFS[1]: Successfully opened standby log 4:
'/u01/app/oracle/oradata/dgfdb/SRL01.log'
Thu Dec 27 17:34:32 2007
Fetching gap sequence in thread 1, gap sequence 35-35
Thu Dec 27 17:34:33 2007
Redo Shipping Client Connected as PUBLIC
-- Connected User is Valid
RFS[2]: Assigned to RFS process 1486
RFS[2]: Identified database type as 'physical standby'
RFS[2]: Archived Log:
'/u01/app/oracle/oradata/dgfdb/archive/1_35_633452428.dbf'
```

```
Thu Dec 27 17:34:39 2007
Redo Shipping Client Connected as PUBLIC
-- Connected User is Valid
RFS[3]: Assigned to RFS process 1686
RFS[3]: Identified database type as 'physical standby'
Thu Dec 27 17:34:39 2007
Redo Shipping Client Connected as PUBLIC
-- Connected User is Valid
RFS[4]: Assigned to RFS process 1688
RFS[4]: Identified database type as 'physical standby'
RFS[4]: Archived Log:
'/u01/app/oracle/oradata/dgfdb/archive/1_36_633452428.dbf'
Thu Dec 27 17:35:03 2007
Media Recovery Log /u01/app/oracle/oradata/dgfdb/archive/1_35_633452428.dbf
Media Recovery Waiting for thread 1 sequence 37 (in transit)
```

References

Oracle® Data Guard Concepts and Administration
10g Release 2 (10.2)
Part Number B14239-04
Chapter 12 Data Guard Scenarios
Chapter 12.6 Using a Physical Standby Database for Read/Write Testing and Reporting

End of the Procedure