

Active Data Guard

In this section I will discuss the following

- Physical Standby - open read-only
- Snapshot Standby - QA and Testing
- Real Application Testing (RAT)
- Active Data Guard

The read-only physical standby database can be used to offload query from the primary database, you can use select statements and complex queries against this this database freeing up resources on the primary database, also remember that the standby database could be localized thus reducing network latency. While the standby is open read-only, the following operations are disallowed

- Any Data Manipulation Language (DML) except for select statements
- Any Data Definition Language (DDL)
- Access of local sequences
- DMLs on local temporary tables

The difference between a read-only and active Data Guard, is that active Data Guard permits both read-only services and disaster recovery while the standby database is open for operation. Which allows for immediate switchover or failover without having to shutdown the standby and put it back in managed recovery mode so it can catch up with the recent redo generated by the primary, also with active Data Guard you will have real-time reporting database.

Read-only Standby

The main reason for a read-only standby is to reduce the resources on the primary, especially if you have a large database for example medical facilities, police departments, manufacturing, these areas generally use volumes of documents. You can also use a read-only database for bug fixing, testing new improved SQL code, temporary change the mode to read-only, perform your testing and then put it back in to recovery mode. If you are using flashback recovery then Data Guard will automatically synchronized the standby database once its been flashed back.

The downside is that once the standby database has been open in read-only mode it is no longer applying redo from the primary and thus becomes out of sync, here are the steps to open in read-only and to put it back in to recovery mode

open in read-only mode	<pre>alter database recover managed standby database cancel; alter database open;</pre>
return to recovery mode	<pre>shutdown immediate; startup mount; alter database recover managed standby database disconnect from session;</pre>
enable real-time apply of redo	<pre>alter database recover managed standby database using current logfile disconnect;</pre>

Snapshot Standby

The difference between a read-only standby and a snapshot standby is that the snapshot standby is fully up dateable. It was possible in Oracle 10g to open a standby database as read-write but as from version 11g you now have the snapshot feature. This new feature makes it simpler to make the standby read-write and to revert to back again with the use of the Broker, also it is advised to use the flashback database feature as it makes life a whole lot simpler.

snapshot standby	<pre>DGMGRL> convert database 'prodldr' to snapshot standby; DGMGRL> show configuration; ## Check that redo is still being sent to the standby select status, sequence#, block# from v\$managed_standby; ## you can double check some tables make sure that you are protected</pre>
revert back	<pre>DGMGRL> convert database 'prodldr' to physical standby; DGMGRL> show configuration; ## shutdown the physical database and restart it mount mode</pre>

As you can see the broker makes light work of this.

Real Application Testing

There are two new features with Oracle 11g

- **Database Replay** - Enables the capture of actual production workloads and the constant, dependable and steadfast execution of a workload in a mirror environment

- **SQL Performance Analyzer** - Provides granular view of the impact of changes in an environment to a SQL statement's execution plan by running statements sequentially before and after the changes have been made.

Database Replay would allow us to capture production workload and test it against a snapshot database before and after changes to that database, the changes could be the following

- Database upgrade, patches, parameter changes
- Configuration changes, such as single instance to RAC, ASM, etc
- Physical changes, such as storage, network, interconnect, bonding/teaming, etc
- Operating system changes, such as hardware migrations, O/S patching, upgrades, etc

SQL performance analyzer would allow us to test

- Schema changes, including the addition, removal or change of an index, partition or materialized view
- Changes in how optimizer stats are collected, the direct application of other SQL performance tuning actions such as creating SQL profiles, histograms, etc.

Firstly lets talk about the database relay feature, once you have converted the physical standby database into a snapshot standby database, you can begin the steps to capture the production workload with database replay.

There are four main components that make up database replay

Workload capture	Workload capture uses binary files (called <i>capture files</i>) to capture any changes received by the Oracle database are tracked and written to these capture files, you can specify where these files are located.
Workload processing	This process is typically run o the snapshot database, the capture files are converted to replay files and and metadata for replaying the workload in the snapshot database. There is no limit to the number of times the converted files can be replayed.
Workload replay	The workload replay uses a replay client program that processes the replay files and submits calls to the database with the same timing and concurrency as in the workload capture system. A calibration tool is provided to assist you in determining how many replay clients you will need to emulate the workload from the original capture system.
Analysis and reporting	Database replay provides extensive reports from simple errors listings to differences in rows returned by DML statements. You can also access AWR reports for detailed comparison between the capture and replay system.

Here are the steps to capture a workload from the primary and replay it several times on the snapshot standby

capture workload and replay it on the snapshot standby	<pre>## First we must configure the Database control on the primary emca -config dbcontrol db -repos recreate start dbconsole ## Create an area for your capture files mkdir /u01/capture_files ## Convert the physical standby to a snapshot standby DMGRL> convert database prodldr to snapshot standby; ## Now configure the database control on the standby database, its easier using database control emcs -config dbcontrol db -repos recreate ## Now start the capture on the primary database 1. Log in to the Database Control on the primary database 2. Click under the software and support link 3. Under Real Application Testing, select Database Replay 4. On the capture workload task line, click the "Go to Task" icon 5. Click Next 6. Provide the area where the capture files will go, perform a test 7. Once the directory object is created click next 8. Set a schedule for your capture and submit your capture job 9. Database Replay will now capture your workload (run any tests that you wish to capture) 10. Return to the Database Control screen and stop the capture 11. When asked to export AWR data click yes. ## At this point you are done with the primary database, copy the capture files to the snapshot standby ## You now need to process the captured files, this can be done on either the primary or the snapshot database but i have ## opt 'ed to do this on the snapshot standby 1. Log in to the Database Control on snapshot database 2. Select the preprocess capture workload task 3. Create the directory object to point to the captured files area 4. click Preprocess workload 5. configure the job settings and verify the replay version, click next 6. set the schedule to run immediately, click next and submit the job ## First we must create a restore point, on the snapshot database sql> create restore point beforereplay guarantee flashback database; sql> shutdown immediate; sql> startup; ## Now we are ready to apply the workload to the snapshot database, at least apply it once without any changes to get a ## baseline 1. Log in to the Database Control on snapshot database</pre>
--	--

```

2. create a directory object to point to the capture files
3. click set up replay
4. Ensure the prerequisites have been met
5. On the restore database requirement, we are using a snapshot standby and flashback database so we do not
   need to restore the database
6. Resolve any references to external system and click continue
7. Determine how many clients you need, use the client program wrc to determine this
8. When the replay sees the clients, it will list them in the table at the bottom, once all accounted for
   click next
9. Click submit on the next page to start the relay and wait until completed
10. Once complete create and save the workload report and the AWR report for your baseline

## Now rewind the snapshot database to the restore point we made earlier
sql> shutdown immediate;
sql> startup mount;
sql> flashback database to restore point before replay;
sql> alter database open resetlogs;
sql> shutdown immediate;
sql> startup;

## Now you can make whatever changes you want, rerun the replay and reproduce your reports and compare, then
repeat this
## whole cycle as many times as you wish

## When you have completely finished its time to revert the snapshot database back to a physical database
sql> drop restore point before replay;

DGMGRL> convert database prod1dr to physical standby;

## Now you should be back as you were.

```

SQL Performance Analyzer is tightly integrated with SQL tuning sets, SQL tuning advisor and SQL lan management functionality. When database replay is running in its client replay mode the SQL Performance Analyzer is capturing detailed statistics and plan information for every DML statement and query executed against the test environment sequentially before and after the changes occur. SQL Performance Analyzer uses these statistics and produces reports that outlines the workload improvements after the changes were made, it also highlights any degrade of performance as a result of the changes made. Use of the SQL Performance Analyzer encompasses five steps

SQL Performance Analyzer

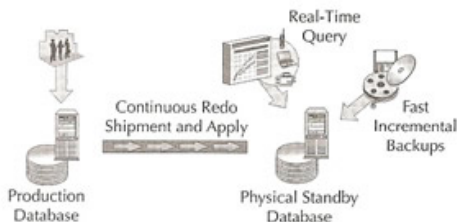
1. Capture the SQL workload you want to analyze with the SQL Performance Analyzer. This would normally be done on the primary using AWR to extract the SQL from the cursor cache into a SQL tuning set. The SQL tuning set is then transferred to the test system where you can analyze it
2. Using the SQL Performance Analyzer, measure you workload's performance prior to any changes.
3. Apply the planned changes to the test environment
4. Repeat step 2 this time with the changes in place
5. Compare the two tuning sets, identifying those that have improved, degraded or stayed the same.

Oracle have a number of whitepapers on [Real Application Testing](#) and they have a dedicate web section as well.

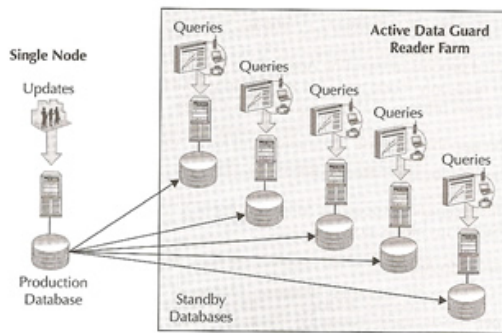
Active Data Guard

We have covered physical standby using read-only and read-write and snapshot standby, all of these variants have one thing in common there is an accepted impact to the recovery time objective (RTO) and potentially to the recovery point objective (RPO).

Active Data Guard allows you to open the physical standby database in read-only mode but what is different is that it will continue to apply redo keeping the standby up to date, this allows you to use this standby as a real-time reporting database or even to backup the primary data, also as a result it does not have any impact on RTO or RPO.



Active Data Guard also supports RAC, which means that you could have a single primary and a Active Data Guard Reader Farm



To enable Active Data Guard, you need to open the physical standby in read-only mode and start redo apply. The Data Guard should be in one of two states prior to enabling Active Data Guard

- The standby is mounted and redo is running
- The standby database has been shutdown cleanly and redo was stopped

To enable Active data guard you can either use SQL Plus or the Broker

Enable Active Data Guard using SQL Plus	<pre>## First stop the redo and open the database as read-only sql> recover managed standby database cancel; sql> alter database open read only; ## Restart the redo sql> recover managed standby database disconnect using current logfile;</pre>
Enable Active Data Guard using the Broker	<pre>## First stop the redo DGMGRL> edit database prodldr set state='apply=off'; ## Open the database as read-only sql> alter database open read only; ## Restart the redo DGMGRL> edit database prodldr set state='apply=on';</pre>

I know what you are think what no option to set to enable Active Data Guard, remember all Active Data Guard is, is a physical standby database that is open read-only with redo being applied, so no special command to turn it on.

That's it for this section, create yourself a test environment and have a play around especially with Real Application Testing, again with everything you test make sue that you have the [logfiles](#) open and see what Oracle is doing in the background, before you know it you will have a very good understanding how Data Guards internally.