ORACLE
DATABASE APPLIANCE

Deploying Oracle Data Guard with Oracle Database Appliance

Applies to Oracle Database Appliance Bare Metal and Virtualized Platform Configurations

ORACLE WHITE PAPER | SEPTEMBER 2016

ORACLE

Introduction

Oracle Database Appliance is a pre-built, pre-tuned, and ready-to-use clustered database system that includes servers, storage, networking, and software in an optimized configuration that makes it easy to deploy, operate, and manage. It is a complete and ideal database platform for small, medium, and large sized implementations and incorporates robust, time-tested Oracle technologies, including the world leading Oracle Database, the best selling Oracle Real Application Clusters (RAC) database option, Oracle Clusterware, and Oracle Automatic Storage Management. By integrating hardware and software, Oracle Database Appliance eliminates the complexities inherent in non-integrated, manually assembled solutions, reducing deployment time from weeks or months to just a few hours, while preventing configuration and setup errors that often result in sub-optimal, hard-to-manage database environments.

Why do I need a standby database environment?

While the Oracle Database Appliance is a highly available system in itself, a standby database environment can provide protection against planned and unplanned downtime as well as against data loss in case the primary database environment becomes unavailable. With the use of proper technology, it is also possible keep the standby database synchronized with the primary database, thereby providing almost transparent continued database operations in the event of planned maintenance such as system patching to problems ranging from user errors to system failures, to disasters. A standby database has therefore always been a key to high availability and protection for any important production system.

Oracle recommends using a separate, dedicated Oracle Database Appliance system to host the standby system for a mission critical production system running on the primary Oracle Database Appliance system.

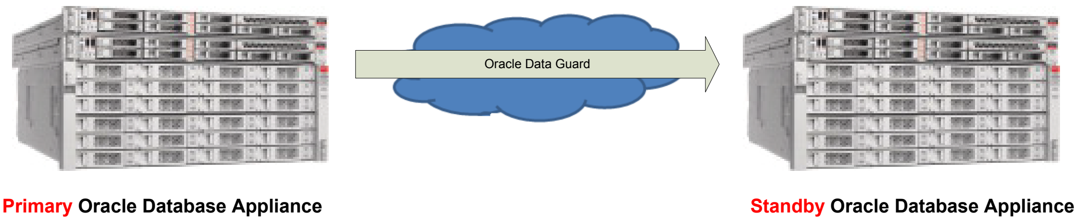
Oracle Data Guard
Primary Oracle Database Appliance
Standby Oracle Database Appliance

Figure 1 - Oracle Data Guard Setup Using Two Oracle Database Appliances

Why Oracle Data Guard?

Oracle Data Guard is the recommended disaster recovery solution to protect databases residing on Oracle Database Appliance against database or cluster failures, data corruptions, disasters, and user errors resulting in “sick” or down database. The tight integration of Oracle Data Guard with Oracle Database provides a unique level of data protection that is impossible to achieve with any other solution. Oracle Data Guard is available to customers as part of Oracle Database Enterprise Edition. It is easy to deploy and provides the management, monitoring, and automation software to create and maintain one or more synchronized copies (standby databases). Oracle Data Guard helps maintain database availability easily when the production database system becomes unavailable due to any reason and also helps minimize downtime during planned maintenance activities by shifting the application workload to the standby environment.

Multiple Benefits of Using Oracle Data Guard

With the use of Oracle Data Guard, the standby database environment does not need to be idle, dark capacity. Instead, the standby database can actively serve many useful purposes. These additional uses greatly increase the overall return on effort and investment.

Migration to Oracle Database Appliance – If you plan to migrate existing databases to Oracle Database Appliance, then Oracle Data Guard enables an easy approach for migration of your databases to Oracle Database Appliance. You can simply setup a Physical Standby database on your Oracle Database Appliance and switchover operations from the old environment to the new environment. This includes migration across certain platforms as well. For example, to migrate your databases currently running on Windows to Oracle Database Appliance, a Linux platform, you may simply setup Oracle Data Guard between the two environments and perform a switchover. This approach to platform migration provides the flexibility to switchback, if for any reason you choose to do so after testing. Refer to My Oracle Support (MOS) note 413484.1 Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration, for more information about platform migration using Oracle Data Guard.

Note: Oracle Data Guard also allows you to migrate across database versions using a transient logical standby database.

Disaster Recovery – Oracle Data Guard physical standby database provides an ideal solution for disaster protection. Disaster scenarios vary from burst water or steam pipes, fire, hurricanes, vandalism, to earthquakes, floods, and acts of terrorism. Oracle Data Guard Physical Standby Database maintains a block-for-block copy of the production database. In the event the primary environment becomes unavailable due to any reason, the standby environment can be quickly activated to maintain continued database availability for your applications.

High Availability – The standby database can also be useful in maintaining availability during planned and unplanned outages and downtimes. Such events may include configuration changes, hardware replacements, and so forth as well as data corruptions, failures resulting from human errors, and other unexpected system component or complete system failures.

Database Rolling Upgrades – The standby database minimizes downtime when patch bundles are applied and changes are made to the primary Oracle Database Appliance. Patches or other maintenance is applied first at the standby database, validated, and then production workload is switched from the primary to the standby system. The only downtime for the databases is the short period of time required to change roles between primary and standby. Please refer to My Oracle Support (MOS) note 1265700.1, Oracle Patch Assurance - Data Guard Standby-First Patch Apply, for more information.

Offloading Workload and Activities – Despite its name, the standby environment does not have to be idle. It can be actively used to maximize the overall return on your investment. With a physical standby database in place, several key activities can be offloaded to the standby environment. These include:

- **Read-Only Workload** – Using Oracle Active Data Guard option, the standby database can be open for read only query workload while being in the standby mode and accepting redo log updates from the primary database. In many cases, offloading read only workload to the standby database can dramatically reduce the production workload, thereby increasing the overall available capacity for the production system.

- **Backups** – Because the Oracle Data Guard physical standby database is a block-for-block copy of the primary database, database backups can be completely offloaded to the standby environment and these backups can be transparently used to restore and recover the primary database in the event of a failure or database loss. Note that if Oracle Active Data Guard option is licensed, then fast incremental backups can be run at the standby database, further adding to the appeal of offloading backups to the standby database.
- **Block Repair** – One of the other benefits of the physical standby database is the ability that it provides to automatically recover from block corruption scenarios. In a primary/standby configuration a corrupt block can be automatically repaired and this operation can be completely transparent to the end user and database administrator. The Block Repair feature is also a part of the Oracle Active Data Guard option.

Snapshot Standby – The Snapshot Standby database is an updatable standby database that provides full data protection for the primary database. It continues to receive redo data from the primary but the apply process is halted while the standby database is open for read/write operations for testing purposes. When testing is complete, a single command reverts the standby database back to its original state, discarding the changes made while it was open in read-write mode and applying the accumulated redo logs to make it synchronize with the current state of primary database.

Best Practices for Setup

This section describes some of the important best practices for setting up Oracle Data Guard on Oracle Database Appliance. For a complete list of general Oracle Data Guard best practices, which also apply to the Oracle Database Appliance environment, please refer to Chapter 8, Configuring Oracle Data Guard, of the Oracle® Database High Availability Best Practices Guide 11g Release 2 (11.2).

- **Match the primary and standby database configuration** -- In order to maintain consistent service levels and to use the primary and standby databases transparently, it is important to match the resources, setup, and configuration of the primary and standby systems as much as possible. Significant differences between the primary and standby database configuration can result in sub-optimal performance and unpredictable behavior when role transitions occur. Specifically, the following recommendations should be considered:
    - *Run Primary and Standby Database on Separate Oracle Database Appliances* - It is recommended that the primary and the standby databases run on separate Oracle Database Appliance units preferably located in a geographically distant location.
    - *Run Primary and Standby Database in Same Configuration* -- Three different database configurations are supported on Oracle Database Appliance; Oracle RAC database, Oracle RAC One, and Single-Instance Enterprise Edition database. The standby database should also be of the same configuration type as the primary database. Thus, if the primary database is configured as an Oracle RAC database, then the standby database should also be configured as an Oracle RAC database.
    - *Size Primary and Standby Instances Similar to Each Other* -- The instances on the primary and standby databases should be configured similar to each other in terms of database parameter settings including memory, CPU, networking, and storage. This helps avoid any unpredictability when the database switch roles. In addition, any operating system configuration customizations should be mirrored in the two environments.

- *Pre-configure Primary and Standby Databases for Role Transition* – The primary and standby databases should be configured so that during role transitions, primary to standby and vice versa, minimal changes are required and necessary. Thus all the database features and customizations implemented on the primary database should be configured on the standby database in advance.

- **Configure Flashback Database on both Primary and Standby Databases** -- The Flashback Database feature enables rapid role transitions and reduces the effort required to re-establish database roles after a transition. As a best practice, Flashback Database should be configured on both the primary and the standby database.

  Note: If Flashback is only used for the purposes of re-instating the Data Guard configuration, it is a best practice to reduce the flashback retention target from the default of 24 hours to 2 hours.

- **Use Dedicated Network for Standby Traffic** -- Oracle Database Appliance comes pre-built with multiple redundant network interfaces. If required, a separate network path can be configured for the standby traffic to minimize any performance impact on the user and application related workload. Note that since Oracle Data Guard needs to transport only the changes made to the primary database from the primary database to the standby database, it does not impose any unnecessary requirements on the network than is needed. Therefore, many deployments of Oracle Data Guard may not require a separate network path to be established for redo log transport between primary and standby. However, some high volume applications or your organizations best practices and standards may require a separate network path for redo log transport. Oracle Database Appliance provides additional network interfaces on each server node for this purpose,. Please refer to MOS note 1422563.1 for additional details on configuring a dedicated network for disaster recovery purposes on Oracle Database Appliance.

- **Consider Offloading Certain Workloads to Standby** -- Oracle Recovery Manager (RMAN) works transparently across the primary and standby databases. Thus database backups taken on the physical standby database can be used for recovering the production database. The standby database should be leveraged to offload backups from the primary database environment. The Oracle Active Data Guard configuration allows for offloading the query workload to the standby environment. Additionally, the physical standby database can also enable transparent block corruption repair.

- **Consider Utilizing Oracle Active Data Guard** – Oracle Active Data Guard allows the standby database to be open for read-only operations while managed recovery (redo transmission from the primary database and its applying on the standby) is concurrently active. This can help distribute the workload from the primary environment to the standby database, increasing the return on investment in the standby database. Note that with Oracle Active Data Guard, it is also possible to use fast incremental backups on the standby database. The fast incremental backups could potentially reduce backup windows from hours to minutes.

Review Oracle Maximum Availability Architecture (MAA) Best practices for Oracle Database - Depending on the deployment and usage of the Data Guard environment, you may find the following additional best practices for Oracle Data Guard useful.

Client Failover Best Practices for Data Guard 12c

http://www.oracle.com/technetwork/database/availability/client-failover-2280805.pdf

Best Practices for Configuring Redo Transport for Active Data Guard 12c

http://www.oracle.com/technetwork/database/availability/broker-12c-transport-config-2082184.pdf

Role Transition Best Practices

http://www.oracle.com/technetwork/database/availability/maa-roletransitionbp-2621582.pdf

Preventing, Detecting, and Repairing Block Corruption - Oracle Database 12c

http://www.oracle.com/technetwork/database/availability/corruption-bestpractices-12c-2141348.pdf

The Maximum Availability Architecture (MAA) Best Practices for Oracle Database 12c are available at

http://www.oracle.com/technetwork/database/features/availability/oracle-database-maa-best-practices-155386.html

Oracle Database Appliance Bare Metal and Virtualized Platform Configurations

Oracle Database Appliance can be configured as a Bare Metal (non-virtualized) platform or as a Virtualized Platform. The Oracle Data Guard Physical Standby setup process outlined in this white paper can be used in both Oracle Database Appliance configurations. On Oracle Database Appliance Virtualized Platform, the configuration steps are executed within the ODA_BASE domain. In addition, Virtual LANs can be used on Oracle Database Appliance Virtualized Platform for configuring a logically separate network for disaster recovery purposes.

Conclusion

Oracle Data Guard enables you to deploy an effective disaster recovery protection strategy right from the time of initial deployment of your Oracle Database Appliance. The physical standby configuration and setup process outlined in this white paper is quick, simple and can be completed without any downtime incurred on the primary database. Most of the standby creation steps are automated using tools such as Oracle Appliance Manager, Database Configuration Assistance (DBCA), RMAN, and Oracle Data Guard and can be used for setting up standby databases on Oracle Database Appliance. However, Appendix A of this white paper provides the complete end to end process for better understanding

Appendix A: Example Setup on Oracle Database Appliance

Sample Environment

The following section describes the primary and standby database environment topologies used in the subsequent Data Guard setup example using Oracle Database Appliance.

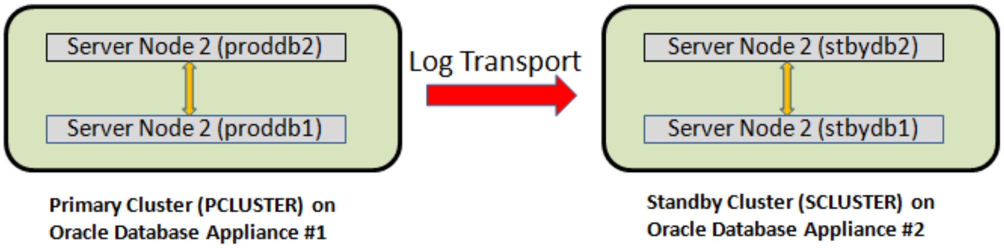
Server Node 2 (proddb2)
Server Node 2 (proddb1)
Log Transport
Server Node 2 (stbydb2)
Server Node 2 (stbydb1)
Primary Cluster (PCLUSTER) on
Oracle Database Appliance #1
Standby Cluster (SCLUSTER) on
Oracle Database Appliance #2

Figure 2 Configuration Topology of Oracle RAC on Oracle Database Appliance

	Primary Oracle Database Appliance		Standby Oracle Database Appliance	
Appliance Name	appliance#1		appliance#2	
Host Names	pnode1	pnode2	snode1	snode2
Cluster Name	PCLUSTER		SCLUSTER	
Database Name	proddb		stbydb	
Instance Name	proddb1	proddb2	stbydb1	stbydb2
SCAN Name and IPs	primary-scan (10.1.27.2, 10.1.27.3)		standby-scan (10.1.27.4, 10.1.27.5)	
Grid Infrastructure Software Installation	/u01/app/12.1.0.2/grid		/u01/app/12.1.0.2/grid	
Oracle Database Software Installation	/u01/app/oracle/product/12.1.0.2/db_home1		/u01/app/oracle/product/12.1.0.2/db_home1	
ARCHIVELOG mode	Yes		Yes	
FORCE LOGGING mode	Yes		Yes	

Table 1 - Example Oracle Database naming conventions

Primary Environment Configuration

1. Create Standby Redo Logs

Standby redo logs host redo data received from the primary database. In advance of the primary standby setup, Oracle recommends that standby redo logs be created on the primary database as well so that it is immediately ready to receive redo data following a switch-over to the standby role.

Create Standby Redo Logs (SRL) on the primary database. Each thread of the standby redo log must have at least one more redo log group than the corresponding thread of the online redo log. For example,

```
$ sqlplus / as sysdba
SQL> alter database add standby logfile thread 1 group 9 size 1024M, group 10 size 1024M, group 11
size 1024M, group 12 size 1024M, group 13 size 1024M;
SQL> alter database add standby logfile thread 2 group 14 size 1024M, group 15 size 1024M, group 16
size 1024M, group 17 size 1024M, group 18 size 1024M;
```

To check the number of online redo logs & their sizes, use the following query.

```
SQL> select group#, thread#, bytes from v$log;
```

Note that the size of the standby redo logs should match the size of the redo logs. The standby redo logs have to be created on the REDO disk group which resides on the solid state disks.

To validate the size of each log file and number of log groups in the standby redo log, use the following query.

```
SQL> select group#, thread#, bytes from v$standby_log;
```

2. Enable archivelog mode on primary database

Archiving is the process of saving and protecting REDO information in the form of archive files before the redo logs of an active database are overwritten in a circular manner. Database created on Oracle Database Appliance have archiving turned on by default. However, it is not mandatory to run your databases in archive log mode.

Verify that the primary database is running in ARCHIVELOG mode.

```
SQL> archive log list
```

If the primary database is not running in ARCHIVELOG mode then enable ARCHIVELOG mode as follows.

Shutdown both instances on Oracle Database Appliance.

```
$ srvctl stop database –d proddb
```

Startup mount one instance in exclusive mode.

```
SQL> startup mount exclusive;
```

Turn on archiving.

```
SQL> alter database archivelog;
```

Shutdown the instance.

```
SQL> shutdown immediate;
```

Restart the database.

```
$ srvctl start database –d proddb
```

3. Enable FORCE LOGGING mode.

Force logging enables you to capture database operations performed with the NOLOGGING attribute. This ensures integrity of your standby database. Verify if FORCE LOGGING is already enabled on your primary database.

```
SQL> select force_logging from v$database;
```

If FORCE LOGGING is not enabled, then enable it using the following commands.

```
SQL> alter database force logging;
```

4. Configure Flashback Database feature

The Oracle Flashback Database feature provide a fast alternative to performing incomplete database recovery. Although using the Flashback Database feature is optional, it can be very useful for faster re-instatement of the old primary database after a failover. Thus, if you do a failover to the standby, and the old primary can be repaired, you do not have to rebuild the old primary database as a standby database but simply flashback and let Oracle Data Guard resynchronize from that point onwards.

Check if the primary database has Flashback Database enabled and if required enable it.

```
SQL> select flashback_on from v$database;
SQL> alter database flashback on;
```

Note that enabling Flashback Database will require additional space consumption in the Fast Recovery Area (RECO Disk Group). The space used by flashback logs can be controlled by setting the parameter DB_FLASHBACK_RETENTION_TARGET to a desired value. This value is specified in minutes. For example,

```
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='*';
```

5. Enable Standby File Management

When the primary database adds or drops a datafile, the corresponding action should also be automatically taken on the standby database. This operation can be enabled using automated standby file management.

```
SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=both sid='*';
```

6. Enable Remote Privileged Login

Ensure that each instance of the primary database is configured with remote login password file. Note that the Oracle Database Appliance deploys the databases with this setting. The initialization parameter REMOTE_LOGIN_PASSWORDFILE must be set to exclusive. If this parameter was reset in your environment and needs to be modified as below, it requires a database restart for it to take effect.

```
$ sqlplus / as sysdba
SQL> show parameter remote_login_passwordfile
SQL> alter system set remote_login_passwordfile='exclusive' scope=spfile sid='*';
```

7. Setup TNS Entries

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that in the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

```
PRODDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = primary-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = proddb)))
```

```
STBYDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = stanby-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = stbydb)))
```

8. Setup Redo Transport Service

The Oracle Data Guard redo transport mechanism uses Oracle Net connections to send the redo between the databases. Redo transport is enabled by setting the LOG_ARCHIVE_DEST_n parameter. For example, the following setup enables log shipping and uses LGWR based transmission in asynchronous mode.

```
SQL> alter system set log_archive_dest_2='SERVICE=stbydb LGWR ASYNC REGISTER
VALID_FOR=(online_logfile,primary_role) REOPEN=60 DB_UNIQUE_NAME=stbydb' scope=both sid='*';
```

More details about redo log transmission options can be found in Oracle Data Guard Concepts and Administration Guide.

9. Setup Fetch Archive Log Server

When the database is in standby role and the primary is unable to send any missing log files, then the standby database can use the FAL_SERVER setting to pull those missing log files. The FAL_SERVER parameter is uses the Oracle Net service name.

```
SQL> alter system set FAL_SERVER=stbydb scope=both sid='*';
```

Standby Environment Configuration

This section describes the steps that must be executed on the standby database. It is assumed that you have set up Oracle Database Appliance system in the standby environment. For setting up Oracle Database Appliance in a Bare Metal or Virtualized Platform configuration please refer to Oracle Database Appliance Setup Poster at http://docs.oracle.com/cd/E22693_01/doc.12/e55694.pdf

10. Setup TNS Entries

Oracle Net Service Names must be configured to enable redo transportation across the databases. Update tnsnames.ora file to include the TNS alias for both primary and standby databases. Note that in the Oracle Database Appliance, the tnsnames.ora file is located in network/admin directory of the Oracle database home.

```
PRODDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST = primary-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = proddb)))

STBYDB =
(DESCRIPTION =
(UT=A)
(ADDRESS = (PROTOCOL = TCP) (HOST = stanby-scan) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = stbydb)))
```

11. Create Static Listener Configuration

As the grid user, create a static listener service on the standby database for Recovery Manager (RMAN) connection during instantiation. Note that the listener home is in the Grid Infrastructure home (/u01/app/12.1.0.2/grid/network/admin)

```
SID_LIST_LISTENER =
    (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = stbydb)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)(SID_NAME = stbydb)))
```

12. Restart Listener

After changes to the listener are made, it must be restarted.

```
grid $> lsnrctl reload listener
```

13. Create Initial Standby Parameter File

Create a parameter file ($ORACLE_HOME/dbs/initstbydb.ora) for the standby database. Ensure that your environment variable for ORACLE_HOME is pointing to the correct home. For example:

```
grid $> echo 'DB_NAME=stbydb' > $ORACLE_HOME/dbs/initstbydb.ora
```

14. Create Password File

During the RMAN duplication process, the auxiliary instance needs to be accessed with remote authentication that requires the creation of the password file.

```
oracle $> orapwd file=/u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/orapwstbydb
password=<primary sysdba passwd>
```

15. Create Audit Directory

Create audit file destination directory on the standby side on both nodes.

```
$ mkdir –p /u01/app/oracle/admin/stbydb/adump
```

16. Create the needed ACFS storage directories on the standby Oracle Database Appliance system

Use the “oakcli create dbstorage” as ‘root’ user command to create ACFS database storage for the standby:

```
# oakcli create dbstorage -db stbydb
```

17. Startup Standby Instance

Startup the standby database instance on first standby host in NOMOUNT state to prepare for instantiation.

```
$> export ORACLE_SID=stbydb
$> sqlplus / as sysdba
SQL> startup nomount
```

18. Validate Network Connectivity

At this stage, Oracle Net should be able to resolve the TNS aliases for both the primary and standby environments from the standby environment.

```
$ tnsping proddb
$ tnsping stbydb
$ sqlplus sys/<password>@//pnode1:1521/proddb as sysdba
```

Instantiate Standby Database

This section outlines the instantiation of the standby database after the setup on the primary and standby environments is complete.

19. Duplicate Database

Using Oracle Recovery Manager (RMAN), the standby database can be created with DUPLICATE DATABASE command. As part of the duplication process, the parameter file, password file, controlfile, and database files are copied over from the primary environment to the standby environment.

The appropriate changes required to the parameter settings for standby operation also need to be specified in the RMAN DUPLICATE DATABASE command. Once RMAN copies over the primary parameter file, the parameters specified in the DUPLICATE DATABASE command are changed accordingly.

As the password file is also copied over, the standby database would have the same password as the primary database.

```
$ export NLS_DATE_FORMAT=”HH24:MI:SS”
$ rman
connect target sys/welcome1@//pnode1:1521/proddb
connect auxiliary sys/welcome1@//snode1:1521/stbydb
run {
allocate channel p1 type disk;
allocate channel p2 type disk;
allocate channel p3 type disk;
allocate channel p4 type disk;
allocate auxiliary channel s1 type disk;
allocate auxiliary channel s2 type disk;
allocate auxiliary channel s3 type disk;
allocate auxiliary channel s4 type disk;
duplicate target database for standby from active database using backupset
dorecover
spfile
parameter_value_convert='/proddb','/stbydb',’PRODB’,’STBYDB’
set db_unique_name = 'stbydb'
set cluster_database = 'false'
set audit_file_dest = '/u01/app/oracle/admin/stbydb/adump'
set db_create_file_dest = '/u02/app/oracle/oradata/datastore/.ACFS/snaps/stbydb'
set log_archive_config="DG_CONFIG=(proddb,stbydb)"
set diagnostic_dest="/u01/app/oracle"
set db_recovery_file_dest="/u01/app/oracle/fast_recovery_area/datastore"
set db_create_online_log_dest_1="/u01/app/oracle/oradata/datastore/stbydb"
set log_archive_dest_2 = 'service=proddb  lgwr async  register  valid_for=(online_logfiles, primary_role)’ db_unique_name=proddb’
set remote_listener = 'standby-scan:1521'
set fal_server='proddb'
set "_cluster_flash_cache_slave_file"=""
set db_flash_cache_file="/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/stbydb/flash1"
nofilenamecheck;
}
```

20. Enable Flashback Database

Enable Flashback Database on the standby database and adjust retention as required. For example,

```
SQL> alter database flashback on;
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=120;
```

21. Start Managed Recovery Mode

Start managed recovery on the standby database in real-time mode as follows:

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

Note: The ‘USING CURRENT CONTROLFILE’ clause is deprecated in Oracle Release 12c. Real-time apply is the default behavior in Oracle Release 12c.

With real-time apply, redo apply services can apply redo log to the standby database as soon as it is received without having to wait for the current standby redo log to be archived. This results in faster failover and switchover times because the standby redo log files have already been applied by the time a failover or switchover occurs.

Post Instantiation Steps

The following steps are performed after the standby instantiation has been completed.

22. Register standby database with Oracle Clusterware.

Make sure that the ORACLE_HOME environment variable is set correctly. Register the standby database with Oracle Clusterware as single instance to run from one node of the cluster.

For a non-Active Data Guard configuration,

```
$ srvctl add database –d stbydb –o $ORACLE_HOME –p
"/u02/app/oracle/oradata/datastore/stbydb/spfilestbydb.ora" –r physical_standby –s mount -c SINGLE
-x snode1
```

For a Active Data Guard configuration,

```
$ srvctl add database –d stbydb –o $ORACLE_HOME –p
"/u02/app/oracle/oradata/datastore/stbydb/spfilestbydb.ora" –r physical_standby –s 'read only' -c
SINGLE -x snode1
```

1. Convert the standby database to Oracle RAC

This step is optional. At this stage the standby database is configured as a single instance database. If the primary database was RAC database, the standby database can also be converted into RAC standby. Appendix B provides information on using the *rconfig* tool to convert the single instance database to RAC standby database.

2. Setup Data Guard Broker Configuration

This step is optional. Creating a Data Guard Broker configuration enables easier management of the entire Data Guard environment as a single entity. It provides management, maintenance and monitoring capabilities that can be used both locally and remotely. Appendix C provides more information on setting up Data Guard Broker configuration.

3. Setup Dedicated DR Network

This step is optional. The Redo Transport Services can be configured to use a dedicated network. A dedicated network channel can help in improving the performance of redo transmission especially when the application network traffic consumes most of available bandwidth on the public network. Please refer to MOS note 1451810.1, Configuring Dedicated Disaster Recovery Network on Oracle Database Appliance, for more information on setting up a dedicated network channel for Data Guard Redo Transport.

4. Verify Configuration and Setup

On the standby database internal data dictionary views can be used to verify standby database operations.

```
$ srvctl config database -d stbydb

SQL> select database_role, switchover_status from v$database;

SQL> select thread#, sequence#, applied from v$archived_log order by sequence#;
```

Appendix B: Converting Single Instance Databases to Oracle RAC

You can use the *rconfig* command line utility to convert a single-instance database to an Oracle RAC database, or to convert it to an Oracle RAC One Node.

To use this feature, complete the following steps:

Create Configuration XML File

A sample of the configuration XML file to be saved as convert.xml is shown below. You may modify this file as required for your system.

The sample XML files are in $ORACLE_HOME/assistants/rconfig/sampleXML directory.

Note: Set the convert option Convert verify="ONLY" initially to perform a test conversion to ensure that a conversion can be completed successfully.

```
<?xml version="1.0" encoding="UTF-8"?>
<n:RConfig xmlns:n="http://www.oracle.com/rconfig"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://www.oracle.com/rconfig rconfig.xsd">
    <n:ConvertToRAC>
        <n:Convert verify="YES">
              <n:SourceDBHome>/u01/app/oracle/product/12.1.0.2/dbhome_1</n:SourceDBHome>
              <n:TargetDBHome>/u01/app/oracle/product/12.1.0.2/dbhome_1</n:TargetDBHome>
              <n:SourceDBInfo SID="stbydb">
                <n:Credentials>
                  <n:User>sys</n:User>
                  <n:Password>welcome1</n:Password>
                  <n:Role>sysdba</n:Role>
                </n:Credentials>
              </n:SourceDBInfo>
              <n:NodeList>
                <n:Node name="snode1"/>
                <n:Node name="snode2"/>
              </n:NodeList>
              <n:InstancePrefix>stbydb</n:InstancePrefix>
              <n:SharedStorage type="CFS">
                 <n:TargetDatabaseArea></n:TargetDatabaseArea>
                 <n:TargetFlashRecoveryArea></n:TargetFlashRecoveryArea>
              </n:SharedStorage>
        </n:Convert>
    </n:ConvertToRAC>
</n:RConfig>
```

Alter the Database Flash Cache parameter as below

SQL> alter system reset db_flash_cache_file sid='*';

Run rconfig Tool

When you have completed making changes, save the file. Run the following command on the standby database. The *convert.xml* is the name of the XML input file you configured above.

$ rconfig convert.xml

Alter the Database Flash Cache parameter as below

```
SQL> alter system set
db_flash_cache_file='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/stbydb/flash1' sid='stbydb1'
scope=spfile;
SQL> alter system set
db_flash_cache_file='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/stbydb/flash2' sid='stbydb2'
scope=spfile;
SQL> alter system set
"_cluster_flash_cache_slave_file"='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/stbydb/flash2'
sid='stbydb1' scope=spfile;
SQL> alter system set
"_cluster_flash_cache_slave_file"='/u02/app/oracle/oradata/flashdata/.ACFS/snaps/flashcache/stbydb/flash1'
sid='stbydb2' scope=spfile;
```

Update Cluster Ready Services Resource

The Cluster Ready Services (CRS) resource must be updated for the converted database.

```
$ srvctl modify database -d stbydb -r physical_standby -s mount
```

Restart the standby database

```
$ srvctl stop database –d stbydb
$ srvctl start database –d stbydb
```

Validate Configuration

Validate the configuration of standby database.

```
$ srvctl config database -d stbydb
```

Appendix C: Creating Data Guard Broker Configuration

This section outlines the process of Oracle Data Guard Broker configuration.

Configure listeners for static registration

Configure listeners for static registration of all the instances of primary & standby databases. In the Oracle Database Appliance, the listeners are running from the Grid Infrastructure home. An example of static registration for a RAC primary & standby configuration:

On node pnode1:

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = proddb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = proddb1)))
```

On node pnode2:

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = proddb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = proddb2)))
```

On node snode1:

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC =
    (GLOBAL_DBNAME = stbydb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = stbydb1)))
```

On node snode2:

```
SID_LIST_LISTENER =
    (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = stbydb_DGMGRL)
    (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
    (SID_NAME = stbydb2)))
```

Configure Broker Configuration Files

Configure location of broker configuration files at both primary and standby databases.

On node pnode1:

```
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/stbydb/broker/dr2.dat '
SCOPE=BOTH SID='*';
```

On node snode1:

```
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE1='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET
DG_BROKER_CONFIG_FILE2='/u02/app/oracle/oradata/datastore/stbydb/broker/dr1.dat'
SCOPE=BOTH SID='*';
```

Enable Data Guard Broker

Enable Data Guard Broker on both primary and standby databases.

On node pnode1:

```
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH SID='*';
```

On node snode1:

```
SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH SID='*';
```

Create Broker Configuration

Create the broker configuration on the primary using the DB_UNIQUE_NAME of the primary database and its corresponding TNS alias.

```
DGMGRL> connect sys/welcome1;
DGMGRL> CREATE CONFIGURATION 'ODADGConfig' AS
> PRIMARY DATABASE IS 'PRODDB'
> CONNECT IDENTIFIER is PRODDB;
```

Add Standby Database to Data Guard Broker Configuration

Add standby database to the configuration using the DB_UNIQUE_NAME of the standby database.

```
DGMGRL> ADD DATABASE 'STBYDB' AS CONNECT IDENTIFIER IS STBYDB;
```

Enable Configuration

Enable Data Guard Broker configuration as follows.

```
DGMGRL> ENABLE CONFIGURATION;
```

Check configuration

Run the following command to verify the established configuration.

```
DGMGRL> show configuration;
DGMGRL> show database verbose prim;
DGMGRL> show instance verbose prim1;
DGMGRL> show instance verbose prim2;
DGMGRL> show database verbose stbydb;
DGMGRL> show instance verbose stbydb1;
DGMGRL> show instance verbose stbydb2;
```

References

Oracle Database Appliance Website on OTN

http://www.oracle.com/technetwork/server-storage/engineered-systems/database-appliance/index.html

Oracle Real Application Clusters Website on OTN

http://www.oracle.com/technetwork/database/clustering/overview/index.html

Oracle Clusterware Website on OTN

http://www.oracle.com/technetwork/database/clusterware/overview/index.html

Oracle Data Guard Website on OTN

http://www.oracle.com/technetwork/database/features/availability/dataguardoverview-083155.html

Oracle Maximum Availability Architecture

http://www.oracle.com/technetwork/database/features/availability/oracle-database-maa-best-practices-155386.html

Oracle Data Guard Concepts and Administration 12c Release 1

http://docs.oracle.com/database/121/SBYDB/toc.htm

My Oracle Support (MOS) Note 1075908.1

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1075908.1

Oracle Database High Availability Website on OTN

http://www.oracle.com/technetwork/database/features/availability/index.html

Oracle® Database High Availability Best Practices 12c Release 1

http://docs.oracle.com/database/121/HABPT/toc.htm

ORACLE

DATABASE APPLIANCE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Hardware and Software, Engineered to Work Together

White Paper Title: Deploying Oracle Dataguard with Oracle Database Appliance
September 2016
Authors: Ramasubramanian Athmanathan, Ravi Sharma, Ravinder Ransi
Contributing Authors: Oracle RAC Pack