Rapid Oracle RAC Standby Deployment: Oracle Database 11*g* Release 2

Oracle Maximum Availability Architecture White Paper
December 2009

Maximum Availability Architecture

Oracle Best Practices For High Availability

ORACLE

Introduction 1
Initial Preparation and Setup 2
Primary Cluster 2
Standby Cluster 3
Process Overview 4
Deployment Example 4
Create the Primary Database 5
Collect Primary Database Information and Prepare the Standby Database 5
Use DBCA to Create an Oracle RAC Database on the Standby Cluster 6
Update Network Configuration Files 9
Prepare the Standby Cluster 12
Create the Oracle Data Guard Broker Configuration 15
Conclusion 18
References 18

Introduction

The rapid Oracle RAC standby deployment process introduces a way to quickly and easily create Oracle RAC physical standby databases using Oracle Recovery Manager (RMAN), Oracle Database Configuration Assistant (DBCA), and Oracle Data Guard broker to automate most of the otherwise manual creation and configuration steps. Automation of configuration steps across multiple cluster systems can eliminate many potential sources of user error and significantly reduce the time required to create an Oracle RAC standby database from an existing Oracle RAC primary database. Additionally, when you are creating a new Data Guard configuration from scratch, some of the standby configuration steps can be performed in parallel with the steps to create the initial primary database so that the total time required to deploy the complete configuration is significantly shortened. This document is part of a series of Oracle Maximum Availability Architecture (MAA) [1] white papers that provide best practices blueprints based on proven Oracle technologies and recommendations that remove complexity from the design of optimal high availability solutions.

The example described in this document is based on an Oracle Database 11*g* Release 2 (11.2) administrator-managed configuration; however, the process also works with Oracle Database 11*g* Release 1 (11.1) configurations.

The example deployment uses Oracle Database 11*g* Release 2 software. Changes required when using Oracle Database 11*g* Release 1 software are noted in callouts.

Oracle Database 11g Release 2 introduces the concept of policy-managed databases using server pools, where Oracle Clusterware allocates and automatically reassigns capacity based on policies you define. The rapid Oracle RAC standby deployment process is designed to work with both Oracle Database 11g Release 1 and Oracle Database 11g Release 2 databases and hence supports only administrator-managed databases that are not part of a policy-managed server pool (the traditional configuration used by databases up through Oracle Database 11g Release 1). For more information on configuring Oracle RAC databases with Oracle Data Guard, refer to *Oracle Data Guard Concepts and Administration* Appendix D.

Initial Preparation and Setup

The rapid Oracle RAC standby deployment process works with any traditional administrator managed Oracle RAC configuration that can be created using DBCA. It works best when the primary and standby databases are configured similarly on similar hardware. The example configuration used in this document consists of two Linux clusters that host administrator managed primary and standby databases and an optional standalone system used to host a Data Guard observer process if fast-start failover is configured. When used in a production environment, the steps described in this document should be augmented with any necessary additional best practices.

Primary Cluster

The instructions given in this document assume the following about the primary cluster:

- Primary system names are dg1, dg2, and dg3 and they form cluster dg-cl1.
- An OCFS2 shared file system accessible to the primary cluster systems is mounted as /shrd_dev.

  While the example documented in this paper uses OCFS, the same procedure can be used to create a RAC standby database that uses Automatic Storage Management (ASM).

- Oracle Clusterware 11*g* Release 2 (11.2) is installed and configured on a private disk on each primary system with home directory /private2/grid.
- Oracle Database 11*g* Release 2 (11.2) is installed on a private disk on each primary system with home directory /private3/oracle/product/11.2.0/db_1.
- ORACLE_BASE is set to /private3/oracle.
- SCAN is dg-cl1-svip with default SCAN listener port 1521. The local listener on each system is LISTENER with default port 1521.

  The SCAN and SCAN listener are new Oracle Database 11*g* Release 2 features.

- The administrator managed primary database is called "east.us.oracle.com" with instances "east1" on dg1, "east2" on dg2, and "east3" on dg3.
- The initialization parameters DB_NAME and DB_UNIQUE_NAME are both set to "east".
- ARCHIVELOG mode, FORCE LOGGING mode, and Oracle Flashback Database are enabled on the primary database.

  FORCE LOGGING mode protects against unlogged direct writes in the primary database that otherwise would not be propagated to the standby database.

- A standby redo log with an appropriate number of standby redo log groups for each primary database thread has been added to the primary database.
- The primary database data files, online log files, password file, server parameter file, and one control file are located in /shrd_dev/oradata/east. The primary database archive, standby redo, and flashback log files, and a second control file are located in a fast recovery area /shrd_dev/`fast_recovery_`area.

  For Oracle Database 11*g* Release 1, the *fast* recovery area is referred to as the *flash* recovery area.

Standby Cluster

The instructions given in this document assume the following about the standby cluster:

- Standby system names are dg7, dg8, and dg9 and they form cluster dg-cl2.
- An OCFS2 shared file system accessible to the standby cluster systems is mounted as /shrd_dev.

  While the example documented in this paper uses OCFS, the same procedure can be used to create a RAC standby database that uses Automatic Storage Management (ASM).

- Oracle Clusterware 11*g* Release 2 (11.2) is installed and configured on a private disk on each standby system with home directory /private2/grid.
- Oracle Database 11*g* Release 2 (11.2) is installed on a private disk on each standby system with home directory /private3/oracle/product/11.2.0/db_1.
- ORACLE_BASE is set to /private3/oracle.
- SCAN is dg-cl2-svip with default SCAN listener port 1521. The local listener on each system is LISTENER with default port 1521.

  The SCAN and SCAN listener are new Oracle Database 11*g* Release 2 features.

- The administrator managed standby database will be called "west.us.oracle.com" with instances "west1" on dg7, "west2" on dg8, and "west3" on dg9.
- The standby database data files, online log files, password file, server parameter file, and one control file will be located in /shrd_dev/oradata/west. The standby database archive, standby redo, and flashback log files, and a second control file will be located in a fast recovery area /shrd_dev/fast_recovery_area.

  For Oracle Database 11*g* Release 1, the *fast* recovery area is referred to as the *flash* recovery area.

Process Overview

The following steps present an overview of the process used to create the Oracle RAC standby database:

- If the Data Guard configuration is being created from scratch and a primary database does not yet exist, create the initial Oracle RAC primary database on the primary cluster using DBCA.
- Collect primary database information needed to create the standby database, create needed directories on the standby cluster, and (if necessary) update primary database initialization parameters to enable future role transitions.
- Use DBCA to create an Oracle RAC database on the standby cluster that is configured with parameters and structure similar to the primary database and that uses the standby database name (west.us.oracle.com). Initially, both the DB_NAME and DB_UNIQUE_NAME initialization parameters will be set to the same value (west). Creating the initial Oracle RAC database on the standby cluster can be done at the same time the primary database is created on the primary cluster if there is no initial existing primary database.
- Update the network configuration files on both clusters (and on the observer system if fast-start failover will be used) to ensure that the network service names and static services required for use with Oracle Data Guard are correctly configured.
- Prepare the standby database cluster for standby database creation by:
  - shutting down the Oracle RAC database on the standby cluster.
  - deleting the associated data files, control files, online redo log files, and archived log files.
  - changing the value of the DB_NAME initialization parameter in the standby database server parameter file from its initial value ("west") to match that of the primary database ("east").
- Use RMAN to create the standby database data files and standby control file from the primary database and to replace the existing standby database password file with a copy of the primary database password file.
- Enable Oracle Flashback Database for the standby database.
- Create an Oracle Data Guard broker configuration.
- Configure an observer process and enable fast-start failover if desired.

Deployment Example

The following example uses Oracle Database 11*g* Release 2 (11.2); however the approach also can be used with Oracle Database 11*g* Release1 (11.1).

Create the Primary Database

If this is a new deployment, use DBCA to create the primary database. Ensure that ARCHIVELOG mode, FORCE LOGGING mode and Oracle Flashback Database are enabled for the primary database and that a standby redo log is added.

Each thread of the standby redo log must have at least one more redo log group than the corresponding thread of the redo log at the redo source database.

For this example configuration, the standby redo log was configured to use multiple standby redo log file groups in the fast recovery area by issuing the following commands while connected to the primary database as SYSDBA:

```
SQL> alter database add standby logfile thread 1 group 7 size 500M,
  2  group 8 size 500M, group 9 size 500M;
Database altered.
SQL> alter database add standby logfile thread 2 group 10 size 500M,
  2  group 11 size 500M, group 12 size 500M;
Database altered.
SQL> alter database add standby logfile thread 3 group 13 size 500M,
  2  group 14 size 500M, group 15 size 500M;
Database altered.
```

It is also a best practice to enable automatic standby file management at the time the primary is created by issuing the following SQL command while connected as SYSDBA:

```
SQL> alter system set standby_file_management=AUTO;
System altered.
```

Collect Primary Database Information and Prepare the Standby Database

Connect to the primary database as SYSDBA and issue a SQL SHOW PARAMETERS command to identify any primary-specific parameters with values (for example, DB_FILE_NAME_CONVERT, LOG_FILE_NAME_CONVERT, or parameters containing a file path or derived from DB_NAME or DB_UNIQUE_NAME) that will require different values to be specified when created on the standby database.

Based on the collected primary database initialization parameter information, create any needed directories on the standby cluster. Then use the srvctl utility to verify that the standby SCAN and local listeners are running by issuing the following commands from one of the standby systems:

```
$ srvctl status scan
$ srvctl status listener
```

For Oracle Database 11*g* Release 1 there is no SCAN listener and you must issue the command

```
$ lsnrctl status listener
```

on each standby system to determine the status of the local listeners.

Before proceeding to the next step, all Oracle software installations and other previously noted initial preparation and setup steps must be completed.

Use DBCA to Create an Oracle RAC Database on the Standby Cluster

If you are creating a deployment from scratch, you can perform this step in parallel with creating the primary database. The goal is to create an Oracle RAC database on the standby cluster with a configuration similar to that of the primary database. DBCA automatically takes care of most network, Oracle Clusterware, Automatic Storage Management (ASM), and other configuration details for this database. Because the data files and control files will be replaced later with copies from the primary database, they do not need to match those used on the primary (so it is fine to use smaller files such as those in the General Purpose seed database to make the database creation process faster). However, the parameter values in the standby server parameter file should parallel those used for the primary database.

Start DBCA

From the command line on the first standby system, issue appropriate operating system-specific commands to configure the database-related environment variables and start DBCA. For the example Linux deployment, issue commands on system dg7 similar to the following:

```
$ setenv ORACLE_BASE /private3/oracle
$ setenv ORACLE_HOME /private3/oracle/product/11.2.0/db_1
$ setenv PATH ${ORACLE_HOME}/bin:${PATH}
$ dbca
```

Initial DBCA Wizard Steps

For steps 1 and 2 shown in the initial DBCA screens, select **Oracle Real Application Clusters database**, **Create a Database**, and **General Purpose or Transaction Processing**. For DBCA step 3, specify Admin-Managed and enter the appropriate database and cluster information.

All RAC databases prior to Oracle Database 11*g* Release 2 are Admin-Managed (where each instance is statically configured to run on a specific cluster node). The Database Identification panel displayed when using DBCA for database releases prior to 11*g* Release 2 will not have radio buttons to select the database Configuration Type.

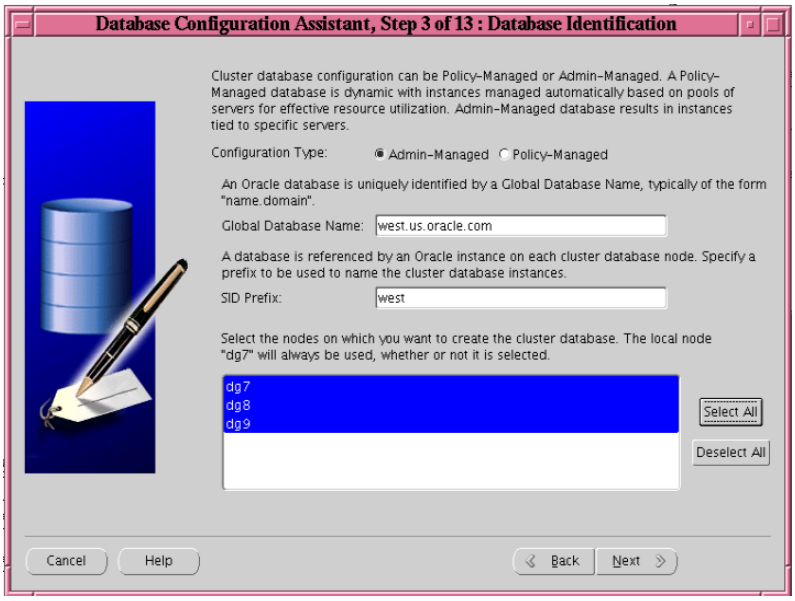

Figure 1: DBCA Step 3, Database Identification

Enter appropriate values for DBCA step 4 and step 5. Use the same SYS user password as on the primary database. Note that the standby database password file will later be replaced with a copy of the primary database password file, so any values entered in step 5 will be superseded later.

Database File Locations

For DBCA step 6, specify file locations that parallel the primary database file locations.

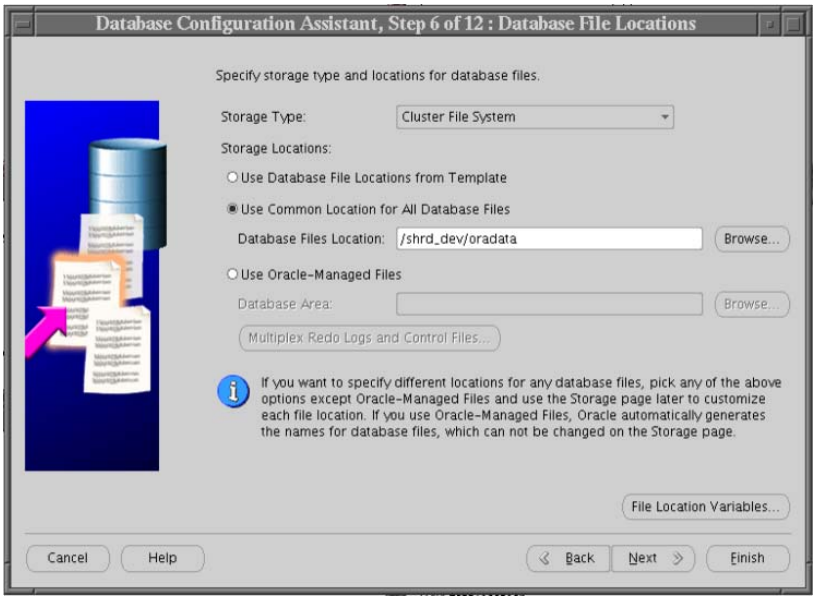

Figure 2: DBCA Step 6, Database File Locations

Although it is possible to use locations on the standby database that do not parallel the locations on the primary database if appropriate file name conversion parameters are specified later in DBCA step 9, it is easier to manage and configure the standby cluster if it is configured as similarly as possible to the primary cluster.

Recovery Configuration

For DBCA step 7, specify the location and size of the fast recovery area and enable archiving. For the example configuration, the specified values match the values used for the primary database.

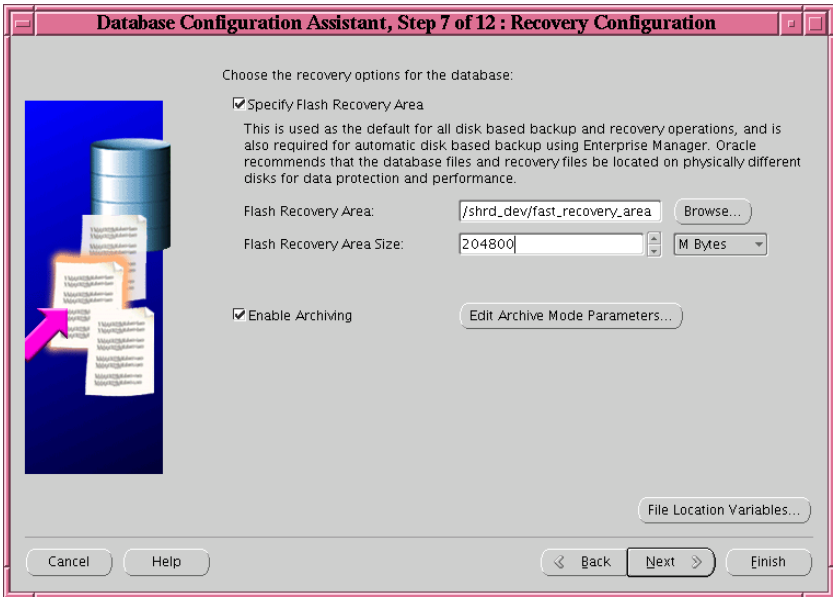

Figure 3: DBCA Step 7, Recovery Configuration

Initialization Parameters

Accept the default value for step 8 and proceed to step 9 to configure initialization parameters. Enter appropriate information for memory use, click **All Initialization Parameters**, and then click **Show Advanced Parameters**. Specify appropriate values, such as those shown in *Table 1*, for the DB_UNIQUE_NAME and STANDBY_FILE_MANAGEMENT parameters, file name conversion parameters, broker configuration file location parameters, and any other parameters that may need adjustment based on your environment. The broker configuration files must be located on shared storage accessible to all database instances. The example configuration also uses the fast recovery area as a local destination for archived log files because it is an Oracle recommended best practice.

INITIALIZATION PARAMETER	VALUE USED FOR EXAMPLE DEPLOYMENT
DB_FILE_NAME_CONVERT	'east','west','EAST','WEST'
DB_UNIQUE_NAME	west
DG_BROKER_CONFIG_FILE1	/shrd_dev/oradata/{DB_UNIQUE_NAME}/dr1{DB_UNIQUE_NAME}.dat
DG_BROKER_CONFIG_FILE2	/shrd_dev/oradata/{DB_UNIQUE_NAME}/dr2{DB_UNIQUE_NAME}.dat
LOG_ARCHIVE_DEST_1	LOCATION=USE_DB_RECOVERY_FILE_DEST
LOG_FILE_NAME_CONVERT	'east','west','EAST','WEST'
STANDBY_FILE_MANAGEMENT	AUTO

Table 1: Standby Database Initialization Parameters

Remaining DBCA Steps

Accept the default values for DBCA step 10. For step 11, if you plan to deploy multiple standby databases from the same primary database, you may find it useful to generate database creation scripts that could be copied and modified for use on other standby clusters. Click **Finish** and then, on the Summary screen, click **OK** to begin creating the Oracle RAC database (and optional database creation scripts).

Update Network Configuration Files

At this point there are two similarly configured but different Oracle RAC databases, each in the primary role. Before creating the standby database, update the network configuration files on all systems to ensure proper client access to both databases and to create the static service entries used by the Oracle Data Guard observer and broker command line utility, DGMGRL.

There may be release specific changes in the way network files are configured for Oracle RAC databases. Specifically, for Oracle Database 11*g* Release 2 (11.2), the listener.ora network configuration file is now located in the Oracle Clusterware home while the tnsnames.ora file remains in the Oracle Database (or client application) home.

To allow DGMGRL (or Data Guard observer process) to restart instances during role transitions, it is necessary to create a static service entry in the listener.ora file on each primary and standby system. The network configuration files can be sensitive to formatting so using a tool such as the netmgr utility to add the needed entries is recommended. Once a tnsnames.ora file has been updated on one system, it can safely be copied to the appropriate location on all other systems. However, because the listener.ora files differ on each system, it is necessary to update each listener.ora file individually.

Tnsnames.ora Files

Ensure that the following service name entries are included in the tnsnames.ora file used on each database system. DBCA automatically adds a correctly formatted service name entry to the

tnsnames.ora file on each cluster system that hosts an instance of a database at the time it creates the database. Connections through the services EAST or WEST will be load balanced across database instances. If desired, additional service name entries such as that for EAST1 (which will be used later) can be added to allow connections to specific database instances.

For the example configuration, the tnsnames.ora file on each database system is located in the network/admin directory of the database Oracle home directory.

```
EAST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dg-cl1-svip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)(SERVICE_NAME = east.us.oracle.com)
    )
  )
```

The SCAN used in the HOST portion of the ADDRESS clause is new for Oracle Database 11*g* Release 2. This replaces the list of ADDRESS clauses (one for each VIP) and explicit (LOAD_BALANCE = yes) clause required for 11*g* Release 1.

```
EAST1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dg-cl1-svip)(PORT = 1521)
    (CONNECT_DATA =
      (SERVER = DEDICATED)(SERVICE_NAME = east.us.oracle.com)
      (INSTANCE_NAME = east1)
    )
  )
WEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dg-cl2-svip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)(SERVICE_NAME = west.us.oracle.com)
    )
  )
```

The tnsnames.ora file in each client application Oracle home also should contain appropriate copies of these entries. However, if all client access will be through a service (for example, SALES) that is available only when a database is running in a specific role, then just the service entry for that service is required in the tnsnames.ora file used by the client application.

```
SALES =
  (DESCRIPTION =
    (CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = dg-cl1-svip)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = dg-cl2-svip)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)(SERVICE_NAME = sales.us.oracle.com)
    )
  )
```

For Oracle Database 11*g* Release 2, the `-l` option of the `srvctl add service` command can be used to create role-based services. For Oracle Database 11*g* Release 1, however, database triggers must be created for each database to stop and start services in response to STARTUP or DB_ROLE_CHANGE events.

Listener.ora Files

The listener.ora file for the local listener on each database system must be updated (using a tool like netmgr) to include a static service entry to allow the DGMGRL command line interface and optional observer process (configured if fast-start failover is enabled) to automatically restart instances during the course of broker operations. Once changes have been made to the listener.ora files, the listeners must be stopped and restarted to ensure the changes are in effect for all client connections. Section 2.2 Prerequisites of the Oracle Data Guard Broker 11g Release 2 (11.2) manual describes this requirement in more detail. For the example configuration, the listener.ora file located in the Oracle Clusterware home on system dg1 (which hosts the Oracle RAC database instance "east1") is updated to add the following static service entry:

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = east_DGMGRL.us.oracle.com)
      (ORACLE_HOME = /private3/oracle/product/11.2.0/db_1)
      (SID_NAME = east1)
    )
  )
```

For Oracle Database 11*g* Release 2 software, listeners are configured to run in the Oracle Clusterware home. The ORACLE_HOME and PATH environment variables must be set to values corresponding to the Oracle Clusterware home when using netmgr to update the listener.ora file. When later working with software in the database Oracle home, these environment variables must be reset to values corresponding to the Oracle Database home.

The corresponding value for SID_NAME in this entry will differ for each cluster system and "east" will be replaced by "west" for the standby systems.

To stop and restart the local listeners on all systems of each respective cluster, the following commands can be issued once from a primary cluster system and once from a standby cluster system:

```
$ srvctl stop listener
$ srvctl start listener
```

For Oracle Database 11*g* Release 1 software, include the `-n` option followed by a comma-separated list of the systems in each cluster when executing these commands.

Verify the Network is Configured Correctly

After all changes have been made to the network configuration files and the listeners have all been stopped and restarted, verify that clients (such as SQL*Plus) can access the databases from all database systems and from any client systems. If you plan to use Oracle fast-start failover, then add copies of the EAST and WEST service name entries to the tnsnames.ora file in the

Oracle home that will later host the observer process and verify that client connections from this system using these service names are possible to both databases.

Prepare the Standby Cluster

With the basic cluster and network configuration now in place, the next step in the process is to prepare the standby cluster so that it is ready for standby database creation. Shut down the current Oracle RAC database on the standby cluster, delete any files that later will be replaced, and update the server parameter file to change the value of the DB_NAME initialization parameter to match that of the primary database.

Shut Down the Oracle RAC Database on Standby Cluster

From a system on the standby cluster, use the srvctl utility to stop the Oracle RAC database ("west") on the standby cluster:

```
$ srvctl stop database -d west -o abort
```

Delete Existing Data, Control, and Log Files

The existing Oracle RAC database on the standby cluster was created as a placeholder so that DBCA could preconfigure the network and Oracle Clusterware environments for the still-to-be-created Oracle RAC physical standby database. This approach also allows database network access to be verified before the standby database is created. The existing data, control, and log files from the placeholder Oracle RAC database will not be used by the standby database and can now be deleted. For the example configuration, the relevant commands are as follows:

```
$ cd /shrd_dev/oradata/west
$ rm *.dbf
$ rm control01.ctl
$ rm /shrd_dev/fast_recovery_area/west/control02.ctl
$ rm *.log
$ rm -r /shrd_dev/fast_recovery_area/WEST/archivelog/*
```

NOTE: For ASM deployments use the ASMCMD command line interface to delete the data, control, and log files (see, for example, http://download.oracle.com/docs/cd/E11882_01/server.112/e10500/asm_util003.htm - CIHGHEID).

Update the Server Parameter File

The initial values of the DB_NAME and DB_UNIQUE_NAME initialization parameters were used by DBCA to determine default values for other initialization parameters and to identify where to locate many database related files on disk. Both parameters were initially set to a value of "west" to ensure that the directory structures and file names created on the standby cluster would parallel the directory structures and file names that were created using "east" on the primary cluster. However, because the physical standby database will be an exact copy of the primary database, the value for DB_NAME in the standby server parameter file (currently "west") must now be changed to match that of DB_NAME on the primary ("east").

To do this, first create a text initialization parameter file from the existing standby server parameter file. For the example configuration, the commands to do this are as follows:

```
$ cd /shrd_dev/oradata/west
$ sqlplus / as sysdba
Connected to an idle instance.
SQL> create pfile='/shrd_dev/oradata/west/initwest.ora'
  2  from spfile='/shrd_dev/oradata/west/spfilewest.ora';
File created.
SQL> exit
```

Using a text editor of your choice, locate the following line in the text initialization parameter file:

```
*.db_name='west'
```

Change it to the following and save the file:

```
*.db_name='east'
```

The value for DB_UNIQUE_NAME must remain "west" to uniquely identify the standby database. To complete the process, create a new server parameter file from the edited text initialization parameter file and then delete the no longer needed text initialization parameter file as follows:

```
$ sqlplus / as sysdba
Connected to an idle instance.
SQL> create spfile='/shrd_dev/oradata/west/spfilewest.ora'
  2  from pfile='/shrd_dev/oradata/west/initwest.ora';
File created.
SQL> exit
Disconnected
$ rm initwest.ora
```

Create Temporary Service Name Entry for the First Standby Instance

RMAN must be able to connect to an unmounted instance of the standby database to execute the DUPLICATE database command. This requires that the RMAN connection to the standby instance be made using a static listener service. Because a static listener service has already been created on each system for use by the Data Guard broker, you can use the netmgr utility to temporarily add an entry for service name STBY for the first standby instance "west1" to the tnsnames.ora files in the network/admin directories of the database Oracle homes for standby instance "west1" (on system dg7) and primary instance "east1" (on system dg1):

```
STBY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dg7.us.oracle.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)(SERVICE_NAME = west_DGMGRL.us.oracle.com)
    )
  )
```

Use RMAN to Create the Standby Database

The RMAN commands used in this example to complete the rapid Oracle RAC standby deployment process can be used with Oracle Database 11*g* Release 2 (11.2) or Oracle Database 11*g* Release 1 (11.1). An RMAN script file is created, one instance of the standby is started with the nomount option, and then the RMAN script is executed to create the standby database data, control, and log files and to replace the existing password file for the standby database with a copy of the primary database password file. Some additional steps are then performed to clean up the environment and complete the configuration of the standby database.

Create the RMAN Script File

Using a text editor of your choice, create and save a text file named *create_standby.txt* containing the RMAN commands required to create the standby database from the active primary database:

```
run {
        allocate channel prmy1 type disk;
        allocate channel prmy2 type disk;
        allocate channel prmy3 type disk;
        allocate channel prmy4 type disk;
        allocate auxiliary channel stby type disk;
        duplicate target database for standby
          from active database;
     }
```

Start the Auxiliary Instance

Start the first instance ("west1") of the standby Oracle RAC database in nomount mode by issuing the following command from the standby system:

```
$ srvctl start instance -d west -i west1 -o nomount
```

Execute RMAN Commands

Start RMAN and issue the commands required to create the standby database. All RMAN connections to the primary database must be through a service name that always connects to the same instance (in this case "east1") and that is not load balanced over other instances. The commands should look similar to the following

```
$ rman
RMAN> connect target sys/******@east1

connected to target database: EAST (DBID=3664244207)

RMAN> connect auxiliary sys/******@stby

connected to auxiliary database: EAST (not mounted)

RMAN> @create_standby.txt
```

Remove the Temporary Service Name Entry

The STBY service name is no longer needed after the standby is created. Use netmgr to remove this service name entry from the tnsnames.ora files on the systems hosting the first instances of the primary and standby databases.

Enable Oracle Flashback Database

Enabling Oracle Flashback Database is an Oracle recommended best practice. If you plan to enable fast-start failover for the configuration, Oracle Flashback Database must be enabled on both the primary and standby databases.

Oracle Flashback Database should already have been enabled on the primary database as part of the initial setup. To enable Oracle Flashback Database on the newly created standby database, issue the following commands:

```
$ sqlplus sys/******@west as sysdba
SQL> alter database flashback on;
Database altered.
SQL> select flashback_on from v$database;

FLASHBACK_ON
------------------
YES

SQL> exit
```

Start the Remaining Standby Database Instances

Issue the following command to start and mount the remaining standby database instances:

```
$ srvctl start database -d west -o mount
```

Create the Oracle Data Guard Broker Configuration

The primary and standby Oracle RAC databases are now created, but log shipping between the databases and redo apply is not enabled. The Oracle Data Guard broker can simplify these final configuration steps and also optionally enable fast-start failover between the databases.

Create the Broker Configuration

Using SQL*Plus, connect to each database (east and west) as SYSDBA and issue the following command to start the Oracle Data Guard broker:

```
SQL> alter system set dg_broker_start=true scope=both;
```

Then from one of the systems, use DGMGRL to connect to the primary database as a user with SYSDBA privileges and issue commands to create and enable the broker configuration. If, after

waiting a few moments to allow broker operations to complete on all cluster systems, `show configuration` does not display `SUCCESS`, then resolve any issues before continuing.

```
DGMGRL> create configuration config as primary database is east connect identifier
is east;
Configuration "config" created with primary database "east"

DGMGRL> add database west as connect identifier is west;
Database "west" added
DGMGRL> enable configuration;
Enabled.
DGMGRL> show configuration
Configuration – config
  Protection Mode: MaxPerformance
  Databases:
    east - Primary database
    west - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS
```

Update Primary Database Parameters to Support Future Role Transitions

If any primary database initialization parameters (such as those listed in Table 1) need updating to support future role transitions, use the DGMGRL command line interface to make these changes now by issuing commands such as the following:

```
DGMGRL> edit database east set property DbFileNameConvert = 'west, east, WEST, EAST';
Property "dbfilenameconvert" updated
DGMGRL> edit database east set property LogFileNameConvert = 'west, east, WEST, EAST';
Property "logfilenameconvert" updated
DGMGRL> edit database east set property StandbyFileManagement = 'AUTO';
Property "standbyfilemanagement" updated
```

The new values for the static initialization parameters DB_FILE_NAME_CONVERT and LOG_FILE_NAME_CONVERT are set in the primary database server parameter file (spfile), but not yet in memory. Because these parameters are not used while the database remains in the primary role, updating the in memory values is deferred until the next time the primary database is restarted (for example, during planned maintenance or during a role transition).

Note that the Data Guard broker log files and `show` commands will display an informational message to alert the administrator whenever the in memory value of a parameter differs from the value set in the server parameter file by the broker.

Optionally Configure and Enable Fast-Start Failover

To enable fast-start failover, each database must be configured with a standby redo log and have Oracle Flashback Database enabled (as is the case in the example). Optionally, the configuration protection mode can be upgraded to Maximum Availability with SYNC log transport; the example uses the default protection mode of Maximum Performance with ASYNC log transport.

Add the EAST and WEST service name entries for the primary and standby databases to the tnsnames.ora file on the observer system. In a separate window, use DGMGRL to start an observer process. Perform this task while connected as a user with SYSDBA privileges to ensure that the observer process can perform role transitions and reconnect to databases automatically without requiring manual user intervention.

```
DGMGRL> connect sys/******@east
DGMGRL> start observer
```

The window used to start the observer must remain open while the observer is running. Alternatively, on Linux systems, the following command will start an observer that runs in the background:

```
$ dgmgrl sys/oracle@****** "start observer" &
```

NOTE: Enterprise Manager Grid Control also can be used to start an observer. An observer started using Enterprise Manager Grid Control can be configured to automatically restart on another host if the initial host becomes unavailable.

Then, in a different window, use DGMGRL to connect to the primary database as a user with SYSDBA privileges and enable fast-start failover. The initial configuration status may indicate that the configuration is lagging; if this condition does not quickly resolve itself, verify that log transport and apply are working correctly and resolve any issues. If necessary, the status of an individual database can be checked by issuing the DGMGRL command `SHOW DATABASE [VERBOSE] <database name>`.

```
DGMGRL> enable fast_start failover
Enabled.
DGMGRL> show configuration
Configuration – config
  Protection Mode: MaxPerformance
  Databases:
    east - Primary database
    west - (*) Physical standby database
Fast-Start Failover: ENABLED
Configuration Status:
SUCCESS
```

Possible Next Steps

- Refer to the *Oracle Data Guard Broker* manual for more information about:
  - Fast-start failover and how to configure the conditions under which automatic failover to the standby database will occur.
  - DGMGRL commands to manually initiate failover or switchover role transitions.
- Refer to the *Oracle Data Guard Concepts and Administration* manual for more information about:
  - Converting a physical standby into a snapshot standby database.

- Opening a physical standby database for read-only access (and also optionally enabling the Oracle Active Data Guard option).
- Converting an Oracle RAC physical standby database into an Oracle RAC logical standby database.

Conclusion

The rapid Oracle RAC standby deployment process allows you to directly create a multi-instance Oracle RAC standby database from an Oracle RAC primary database without any downtime on the primary and without the intermediate step of first creating a single-instance standby database. In addition, most of the creation and configuration steps are automated by using DBCA, RMAN, and the Oracle Data Guard broker. This automation eliminates the risk of many potential errors during deployment and significantly reduces the time required to create an Oracle RAC physical standby database. When deploying a new configuration from scratch, the creation of the primary database and initial Oracle RAC database on the standby cluster can be performed in parallel.

References

1. Oracle Maximum Availability Architecture Web Site on OTN
   http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm
2. Oracle Data Guard Web Site on OTN
   http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html
3. Oracle Real Application Clusters Web Site on OTN
   http://www.oracle.com/technology/products/database/clustering/index.html
4. Oracle Clusterware Web Site on OTN
   http://www.oracle.com/technology/products/database/clusterware/index.html

Rapid Oracle RAC Standby Deployment:
Oracle Database 11*g* Release 2
December 2009
Author: Laurence Clarke

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

Oracle is committed to developing practices and products that help protect the environment

0109