Is Your DR really available in case of Disaster?

Sreekanth Chintala, Dell Inc

Abstract

Many corporations rely on Data Guard as a primary Disaster Recovery (DR) solution. Once configured properly, switching over to DR site should not be a surprise. However, configuration changes occur over time and chance of maintaining a history of changes that is readily available during the disaster is slim. Validation of a DR configuration involves numerous time consuming steps, yet these steps are essential for a successful switchover. Performing these validations in the middle of a disaster will prolong the system availability. The very purpose of having a DR system is to provide business continuity with as little downtime as possible. How good is it to invest-in or rely-on a DR system that may not meet your business continuity service level agreements?

This paper outlines the usage and advantages of using a utility that can validate a Data Guard configuration and provide a report within a matter of seconds. This paper also briefly discusses the best practices for creating a Physical Standby database and the advantages of leveraging Oracle Enterprise Manager’s extensibility features in creating a Management Plug-in for Data Guard monitoring.

Challenges in a Global Enterprise

In large enterprises, over time, it is possible that the responsibility of managing a database environment may move from DBA to DBA. The person, who configured an environment, may have moved on, and the new owner may not be familiar with the system. Typically, no documentation is maintained on configuration changes. Even if there is documentation, it is usually hard to estimate on the accuracy and authenticity of such documents or there may be many DBAs managing the same system and tracking the changes can be cumbersome and impractical. In essence, in a global enterprise, it is hard to predict the health of a Data Guard environment without performing proper validation.

Proactive vs Reactive

It is a very well known fact that *“Prevention is better than cure”*. How many of us really practice this in life? The reality is we never have time for proactive maintenance and majority of the time we are always fighting the fires reactively. So, we need to find ways to change this behavior through innovation and automation. Performing a manual validation is time consuming and can be error prone. After all “*it is human to err*”. A single mistake or ignoring a critical warning could compromise your system availability down the road. The best solution for avoiding human errors is not to provide the opportunity at all. A utility that can perform the validations and provide a comprehensive report is invaluable.

Data Guard Configuration Verification Utility (DGCVU)

Data Guard Configuration Verification Utility (DGCVU) provides the configuration information in a matter of seconds for your review. Manually performing these validations will take a long time, especially in a RAC environment. As the number of nodes in the RAC increases, the validation time also increases. Moreover, chances of missing one or two critical checks could cost you dearly down the road.

Data Guard

Oracle Data Guard provides a lot of mission critical features such as high availability, data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases which enables primary databases to survive disasters and data corruptions. Data Guard maintains these standby databases as transactionally consistent copies of the primary database. In case the primary database becomes unavailable, Data Guard can switch any available standby database it managers to the primary role, minimizing the downtime associated with the outage.

Data Guard Components

Oracle Data Guard consists of a primary database and a physical or logical standby database. The databases in a Data Guard configuration are connected by Oracle Net and may be dispersed geographically. There are no restrictions on where the databases are located, provided they can communicate with each other. For example, you can have a standby database on the same system as the production database, along with two standby databases on other systems at remote locations. Figure:1 shows different components of a Data Guard environment.



Figure1: Data Guard Components

Primary Database

A Data Guard configuration must contain one primary database that is accessed by most of the applications. The primary database can be either a single-instance Oracle database or an Oracle Real Application Clusters database.

Standby Database

A standby database is a transactionally consistent copy of the primary database. A Data Guard configuration can support up to nine standby databases. Once created, Data Guard automatically maintains each standby database by transmitting redo data from the primary database and then applying the redo to the standby database. Similar to a primary database, a standby database can be either a single-instance Oracle database or an Oracle Real Application Clusters database. A standby database can be either a physical standby database or a logical standby database:

Physical Standby

Physical Standby provides a physically identical copy of the primary database. The database schemas, including indexes, are the same. A physical standby database is kept synchronized with the primary database, through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database. A physical standby database can be used for business purposes other than disaster recovery on a limited basis.

Logical Standby

Logical Standby contains the same logical information as the production database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database through SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executing the SQL statements on the standby database.

A logical standby database can be used for other business purposes in addition to disaster recovery requirements. This allows users to access a logical standby database for queries and reporting purposes at any time. Also, using a logical standby database, you can upgrade Oracle Database software and patch sets with almost no downtime. Thus, a logical standby database can be used concurrently for data protection, reporting, and database upgrades.

Data Guard Broker

The Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. The Data Guard broker logically groups primary and standby databases into a configuration that allows the broker to manage and monitor them together.

You can perform all management operations locally or remotely through the broker’s easy-to-use interfaces: the Data Guard management section of Oracle Enterprise Manager or the Data Guard command-line interface called DGMGRL.

Advantages of using Data Guard broker

There are several advantages in using a Data Guard Broker to manage the configuration.

* Broker manages and monitors the status of the entire configuration. It captures diagnostic information, reports statistics such as the log apply rate, the redo generation rate, and detects problems proactively.
* Enables *Fast Start Failover* operations enhancing high availability. Fast Start Failover is initiating failover to standby automatically upon loss of the primary database without manual intervention.
* Users can manage and monitor all databases in the configuration from any system in the configuration through a client connection.
* Changes to the configuration can be performed through Data Guard Broker command line interface (*dgmgrl*) or through Oracle Enterprise Manager (OEM).
* Performs switchover or failover operations with a single command. All the complexities involved in role reversals are managed by the Data Guard Broker.

Physical Standby Prerequisites and Best Practices

Oracle Data Guard offers lot of flexibility. So the configuration details vary depending on the options chosen. This article covers the prerequisites and best practices of a Physical Standby. The reference architecture used in the examples is as follows:

* Primary database: 2-node RAC
* Physical standby: 2-node RAC
* Storage: Automatic Storage Management (ASM) on Primary and Standby
* Management: Data Guard Broker

Prerequisites and Best Practices for Physical Standby

The following section outlines the best practices that are based on Oracle Maximum Availability [1] guidelines. Please refer to Oracle Data Guard Concepts and Administration [2, 3] for in depth information.

* All the members of a Data Guard configuration must run an Oracle image that is built for the same platform. The operating system running on the primary and standby locations must be the same, but the operating system release does not need to be the same.
* The same release of Oracle Database Enterprise Edition must be installed on the primary database and all standby databases in a Data Guard configuration.
* The standby database can use a different directory structure from the primary database.
* Initialization parameters, *compatible*, *log\_archive\_format* and *log\_archive\_config* need to be same on both primary and standby.
* LOG\_ARCHIVE\_CONFIG initialization parameter need to include the *db\_unique\_name* of all the databases in the configuration.
* The primary database must run in ARCHIVELOG mode and use FORCE LOGGING.
* Set the *db\_unique\_name* initialization parameter for standby database. For example, if primary database is testp, set *db\_unique\_name* for standby as testp\_dr.
* When Oracle Automatic Storage Management (ASM) and Oracle Managed Files (OMF) are used in a Data Guard configuration, set it up symmetrically on the primary and standby database. That is, if any database in the Data Guard configuration uses ASM, OMF, or both, then every database in the configuration should use ASM, OMF, or both, respectively.
* Use *db\_file\_name\_convert* and *log\_file\_name\_convert* initialization parameters if the directory structure on primary and standby are different.
* Set *log\_archive\_max\_processes* to 30.
* Set *remote\_login\_passwordfile* to EXCLUSIVE.
* Set SQLNET.EXPIRE\_TIME=1 in *sqlnet.ora* file.

Pre-requisites for using Data Guard Broker

* Use a server parameter file (SPFILE) to ensure the broker can persistently reconcile values between broker properties and any related initialization parameter values.
* The value of the *dg\_broker\_start* parameter must be set to TRUE.
* For a RAC database, set the *dg\_broker\_config\_file1* and *dg\_broker\_config\_file2* initialization parameters for that database such that they point to the same shared files for all instances of that database. The default values for these parameters will not work. These parameters must specify a raw device, ASM file, or cluster file system file that resolves to the same set of physical files for all RAC instances.
* For RAC database, the START\_OPTIONS for that database must be set to MOUNT in the Oracle Cluster Repository (OCR) using srvctl.
* The LOCAL\_LISTENER initialization parameter on each instance that is part of a Data Guard broker configuration must resolve to a listener address that is reachable by all members of the configuration. LOCAL\_LISTENER initialization parameter need to be setup if a non-default port number (1521) is used in the listener address.
* To enable the Data Guard broker's Command Line Interface (CLI) to restart instances during the course of broker operations, a service with a specific name must be statically registered with the local listener of each instance. The value for the GLOBAL\_DBNAME attribute must be set to a concatenation of <db\_unique\_name>\_<DGMGRL>.<db\_domain>.

For example, in the listener.ora file:

LISTENER = (DESCRIPTION =

(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=host\_name)

(PORT=port\_num))))

SID\_LIST\_LISTENER=(SID\_LIST=(SID\_DESC=(SID\_NAME=sid\_name)

***(GLOBAL\_DBNAME=db\_unique\_name\_DGMGRL.db\_domain)***

(ORACLE\_HOME=oracle\_home)))

* Primary and standby databases need to have same SYS password.

Sample Physical Standby Configuration

The following section shows an example of configuration settings for a typical RAC Data Guard configuration

|  |  |  |
| --- | --- | --- |
| **Property**  | **Primary (2-node RAC)** | **Standby (2-node RAC)** |
| Host names  | gdmsengdb05 gdmsengdb06  | gdmsengdbrdb05 gdmsengdbrdb06  |
| db\_name  | r264s  | r264s  |
| Instance names  | r264s1 and r264s2  | r264s1 and r264s2  |
| db\_unique\_name (Database unique name)  | r264s  | r264s\_dr  |
| db\_domain  | prd.dell.com  | dr.dell.com  |
| Global database name  | r264s.prd.dell.com  | r264s.dr.dell.com  |
| log\_archive\_dest (in ASM) | ARCH\_1  | ARCH\_1 |
| # of online Redologs/thread  | 4 | 4 |
| Size of online Redologs  | 500M | 500M |
| Character Set  | UTF8 | UTF8 |
| TNS alias  | r264s\_prd  | r264s\_dr  |
| ASM diskgroups | DATA\_1, ARCH\_1, FRA\_1  | ARCH\_1 and DATA\_1  |
| Database version  | 10.2.0.3  | 10.2.0.3  |
| Platform  | Linux SLES 10  | Linux SLES 10  |
| Compatible  | 10.2.0  | 10.2.0  |
| Force Logging | YES | YES |
| Standby Redo Log Size | 500M | 500M |
| Standby Redologs/Thread | 5 | 5 |
| Local\_Listener | '(ADDRESS=(PROTOCOL=TCP)(HOST=gridesldb05-vip.prd.dell.com)(PORT=1521))' scope=both sid='**r264s1**''(ADDRESS=(PROTOCOL=TCP)(HOST=gridesldb06-vip.prd.dell.com)(PORT=1521))' scope=both sid='**r264s2**' | '(ADDRESS=(PROTOCOL=TCP)(HOST= gdmsengdbrdb05-vip.prd.dell.com)(PORT=1521))' scope=both sid='**r264s1**''(ADDRESS=(PROTOCOL=TCP)(HOST= gdmsengdbrdb05-vip.prd.dell.com)(PORT=1521))' scope=both sid='**r264s2**' |
| Services | R264s\_online.prd.dell.com | R264sx\_online.prd.dell.com |
| Log\_archive\_config | 'dg\_config=(r264s,r264s\_dr)' | 'dg\_config=(r264s,r264s\_dr)' |
| SQLNET Parameters | Expire\_time=1 | Expire\_time=1 |
| Log\_archive\_max\_processes | 30 | 30 |
| Log\_archive\_format | %t\_%s\_%r.dbf | %t\_%s\_%r.dbf |
| Fal\_client | R264S\_PRD | R264S\_DR |
| Fal\_server | R264S\_DR | R264S\_PRD |
| Dg\_broker\_start | TRUE | TRUE |
| dg\_broker\_config\_file1 | '+DATA\_1/r264s/dr1r264s.dat' | '+DATA\_1/r264s/dr1r264s.dat' |
| dg\_broker\_config\_file2 | '+DATA\_1/r264s/dr2r264s.dat' | '+DATA\_1/r264s/dr2r264s.dat' |
| Standby\_file\_management | AUTO | AUTO |

Validating Data Guard Configuration

Data Guard Configuration Verification Utility (DGCVU) validates the database (Primary or Physical Standby) configuration against the best practices explained earlier. This version of DGCVU doesn’t compare and contrast the similarities between primary and standby databases. However, the reports generated from Primary and Standby can be compared side by side manually.

The following screenshots describe some important features of DGCVU.

1. DGCVU should be run on the database server. It sources the “*.bash\_profile*” or the “*.profile*” of the Oracle user to get environmental information. DGCVU expects ORACLE\_SID and ORACLE\_HOME, ORA\_CRS\_HOME ( in case of RAC) are defined in the profile.
2. High level view of the environment:



* 1. The target database is a 2-node RAC.
	2. DGCVU is being run from node1.
	3. The *Region* is validated against approved list of “*Regions*” that are defined in the script.
	4. The target database is a *Physical Standby*. This information is collected through a *SQLPLUS* connection to the database.
	5. Data Guard Broker is configured.
	6. Data Guard Broker process (DMON) is running on this environment.
1. Operating System information. DGCVU is RAC-aware. The following operating system level checks are performed on all nodes in the cluster provided SSH is enabled and no password is need to make SSH connection.



* 1. OS information and platform are database version are extracted.
	2. The following validations are performed.
		1. *Media Management Layer* is configured for tape backups.
		2. *Automatic Storage Management (ASM)* is configured and ASMinstance is running.
		3. Database is running.
		4. Data Guard Broker is running. This check is only performed if *dg\_broker\_start* initialization parameter is set to *TRUE.*
		5. Media Recovery Process (MRP) is running. This check is only performed if this is a standby database. In 10g, only one of the RAC instances will be the *applying instance.*
		6. SQLNET tuning parameters are configured properly.
		7. GLOBAL\_DBNAME is properly configured in *listener.ora* file. This check is only performed if *dg\_broker\_start* initialization parameter is set to *TRUE.*
		8. TNS entries are properly configured in *tnsnames.ora* file.
		9. TNSPING check validates that the information provided in the *tnsnames.ora* file actually refers to the correct database servers.
	3. After performing OS level checks, applying instance information is provided. This is information is provided only for Physical Standby.



1. Initialization parameter validation. The utility validates some critical initialization parameters. This validation is broken down into three sections. The report lists the parameter, whether the validation is a *Pass* or *Fail*, *Expected Value* and the *Current Value*.
	1. Non-Critical Parameters. As the title suggests, these parameters listed are not really critical for Data Guard to function properly but it is a good idea to validate some important parameters. This is not an exhaustive list but few validations are performed.



* 1. Parameters that should be same between primary and standby. The following parameters need to be the same between primary and a physical standby database.



* 1. Critical Parameters. These parameters need to be set on both primary and standby databases.
		1. db\_domain: This value depends on your deployment standards.
		2. db\_create\_file\_dest: This value need to be set. In this example, it is expected to be an ASM destination.
		3. dg\_broker\_start: The standard is to enable Data Guard Broker.
		4. dg\_broker\_config\_file1, dg\_broker\_config\_file2: Data Guard Broker requires two configuration files to store the configuration information. The recommended standard is to store these files in ASM. The location of the file inside ASM is trivial as long as the destination is ASM.
		5. fal\_client: This parameter refers to the standby that applying the logs. It is not required to set this for Standby if Broker is enabled. Broker dynamically manages the value of this parameter.
		6. fal\_server: This parameter refers to the primary database. It is not required to set this for Standby if Broker is enabled. Broker dynamically manages the value of this parameter.



1. REDOLOG validation. It is important to configure Online Redo Logs (ORL) and Standby Redo Logs (SRL) properly so that the standby keeps up with primary in an efficient manner. *Real Time Apply* ensures the delta between primary and standby to a minimum.



* 1. ORL size is calculated.
	2. SRL size is calculated.
	3. ORL and SRL size are compared. It is required to have same size of ORL and SRLs to ensure Real Time Apply is used.
	4. Number of SRLs are validated against the number of ORLs. The formula is (1 + Number of ORLs/thread) \* Number of Threads.
1. Local Listener Parameters are extracted for each instance and validates the values for PROTOCOL, HOST and PORT.



1. Clusterware information is validated against the set standards. In this example you will see the valid domains, regions that need to be set. START\_OPTIONS is required to set to *mount* when Data Guard Broker is used. It is also important to create all the primary database services on the standby database.



1. Broker Validation: DGCVU extracts the configuration status from Data Guard Broker command line utility “*dgmgrl”*. In the following, broker is configured properly and no errors are reported in the configuration.



Oracle Enterprise Manager(OEM)

In the recent years Oracle Enterprise Manager ( Grid Control) has emerged as an enterprise level monitoring solution. OEM has come a long way from it’s initial humble introduction as graphical user interface for database management to a one stop shop for enterprise level monitoring.

Default Monitoring

Out-of-box, Enterprise Manager can monitor the most common hardware and applications (target types) used in enterprise environments. Because it is not possible to anticipate all possible target types that may exist in your IT environment, Enterprise Manager provides a modular way to extend monitoring capabilities called Management Plug-ins. A Management Plug-in lets you create custom target types that allow you to monitor applications or hardware unique to your enterprise directly from the Enterprise Manager console. You simply deploy the Management Plug-in to Management Agents throughout your enterprise.

Extending Oracle Enterprise Manager

SELECT journal (Quarter 2 2006 and Quarter 3 2007) published articles by Alex Gorbachev “Extending Oracle Enterprise Manager 10g” [5, 6] are some of the best resources on how to create a custom management plug for operating system level scripts and embed them in to your enterprise monitoring. Oracle’s “Extensibility Guide” [6] is also a good reference to understand the internals on extending management plug-ins.

Now that you have a custom script that can validate your Data Guard environment, you can use the extensibility features to extend the default monitoring to alert on your business needs. Using the DGCVU output in its entirety as a new target type can be a bit over whelming. Depending on your organizational needs you can choose what elements can be used to be monitored by OEM.

References

1. Oracle MAA White Paper: <http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10g_RACPrimaryRACPhysicalStandby.pdf>
2. Oracle Data Guard Concepts and Administration 10g Release 2: B14239-04
3. Oracle Data Guard Broker 10g Release 2: B14230-02
4. Oracle Enterprise Manager Extensibility Guide for 10g Release 2: B16246-02
5. Extending Grid Control by Alex Gorbachev: <http://www.oracloid.com/papers/ExtendingOracleEnterpriseManager10g.pdf>
6. Management Plug-ins for 10g Grid Control by Alex Gorbachev: <http://www.ioug.org/private_files.cfm/selectjournal/issues/07q3/SELECT-Q307_Gorbachev-Plug.pdf?redirect=true&file_name=selectjournal/issues/07q3/SELECT-Q307_Gorbachev-Plug.pdf&CFID=6190371&CFTOKEN=80678233>

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Oracle, Data Guard and Enterprise Manager are registered trademarks of Oracle Corporation. Linux is a registered trademark

of Linus Torvalds. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.