

Maximum Availability Architecture of Oracle Servers

J Prabhudas

*Student, Department of Information Technology
VR Siddhartha Engineering College, Vijayawada kanuru, Andhra Pradesh, India*

T Lakshmi Surekha

*Assistant Professor, Department of Information Technology
VR Siddhartha Engineering College, Vijayawada kanuru, Andhra Pradesh, India*

M Ramesh

*Assistant Professor, Department of Information Technology
VR Siddhartha Engineering College, Vijayawada kanuru, Andhra Pradesh, India*

Abstract- The data that is stored to a primary database server from applications has to be backed up to a standby database server for disaster recovery. If the primary database is not available because of network fluctuations, the data from applications cannot be maintained, hence to shift data from primary to standby database server and to switchover from standby to primary server, Data Guard is implemented. Maintaining redundant data with consistency and synchronization will prevent the application data not to lag behind single server failures. If the production (Primary) database is unavailable for any reason, client connections can quickly, and applications transparently, failover to the synchronized replica to restore service by implementing IP failover using keepalived in Redhat servers. Keepalived in Redhat servers (V6.5) will make client connections quickly transfer to one of the replicated Standby server by using Virtual Redundancy Routing Protocol (VRRP) with Priority based system. Aiming to overcome from data loses and to provide maximum availability of transactional data and for disaster recovery Oracle Data Guard technology is used. Data Guard maintains these standby databases as transactional consistent copies of the production database. If the production database becomes unavailable because of network disasters, Data Guard can switch any standby database to the production role.

Keywords – Primary database server, Standby database server, Data Guard, switchover, failover, Keepalived, VRRP, IP Failover.

I. INTRODUCTION

Maximum Availability Architecture (MAA) is the challenging task for every organization to implement. It ensures highest level of data availability for enterprise application servers that are located at different locations. MAA consists of one primary database and one or more synchronized standby or secondary databases that are in consistent state. Hot Standby is nothing but when the Production server is not available, one of the standby database will take over the role of Production so that the applications won't lag and users may not lose any data. The automatic conversion of Standby to Primary is done by configuring data guard. The application connections are transferred from primary server to standby database server when primary is not available, by configuring virtual IP between primary and standby databases. Oracle Data Guard promises high availability, data protection, and recovery from disasters for enterprise data. Oracle Data Guard maintains services that create, maintain, and manages more than one standby database to convert to production databases to survive disasters.

A. Oracle Data Guard Services

Data Guard maintain consistency of primary and standby's by transferring the data that is committed in primary called as Redo data in the form of archived redo logs. Oracle's Data Guard will transmit redo data, applying of redo data, and changes to the database roles by achieving fast start failover.

Redo transport services manage the automated transfer of redo data from the production database to at least one or more standby archival destinations. Resolving archived redo log file gaps due to a network failure. Automatically will find damaged archived redo log files sent from primary on a standby system and automatically retrieve replacement of archived redo logs from production database or standby database.

Redo Apply Services will transmit redo data from the production database is written to the redo logs on the standby database. Apply services automatically apply standby database redo data to maintain consistency with

the primary database. **Redo Apply** technologies used for physical standby database using recovery manager, which applies redo data on the standby database using recovery manager of an Oracle database.

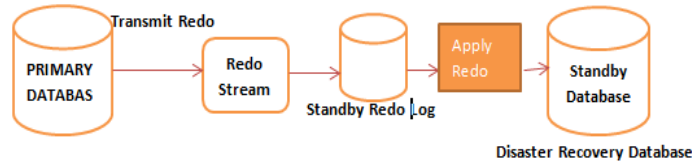


Figure 1. Data Guard Redo Transport Service

B. Role Transitions

An Oracle database functions in two roles: primary or standby. Using Oracle Data Guard, you can convert the role of a database by initiating switchover or fast start failover. A **switchover** is a process of converting role from primary to one of its standby databases. No data will miss by switchover. During a switchover, the primary database changes to a standby role and the standby database changes to the primary role. A **fast start failover** is a process that occurs when primary database is not available. Failover is performed by using observer system if the primary database fails and the failover results in role transition of a standby database to the production role. Oracle Data Guard broker will automatically failovers using DGMGRL command-line interface.

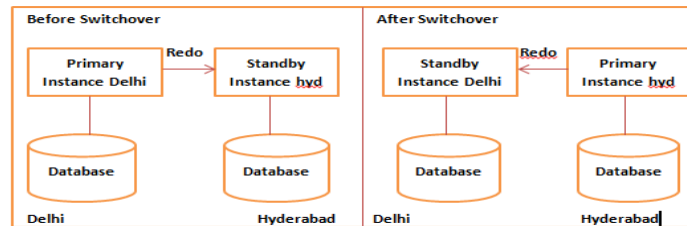


Figure 2. Data Guard Role Transition.

C. Oracle Data Guard Broker

Our main goal is to make Standby Database Server to Primary Database Server when the Primary Database is unavailable in spite of any reason. So for that Fast_Start Failover has to implement and we need third system called Broker or Observer. Fast_start failover is implemented in Maximum Availability Protection Mode since the final system will become Maximum Availability Architecture. Faststart failover allows the broker to automatically fail over to a chosen standby database in the event of loss of the primary database. Once the observer is started, no further user interaction is needed. If each the observer and selected standby database lose connectivity with the primary database for longer than the Amount of seconds specified by the FastStartFailoverThreshold configuration property, the observer will initiate a faststart failover to the standby database. Additionally, the first database will shut down if it perceives a loss of connectivity for an amount longer than FastStartFailoverThreshold seconds, if the FastStartFailoverPmyShutdown configuration property is set to TRUE. When the failover completes, the former primary database is automatically reinstated as a standby database when a connection to it is re-established, if the FastStartFailoverAutoReinstate configuration property is set to TRUE.

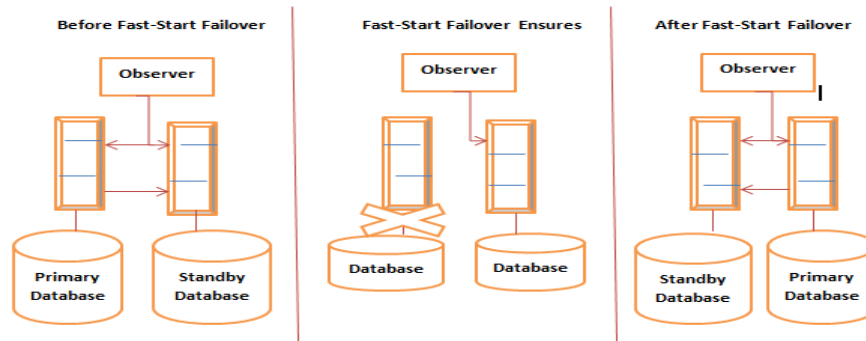


Figure 3. Data Guard Observer that initiates Fast Start Failover

II. IMPLEMENTATION DETAILS

SPFILE is the file that is used for any database to start, the configuration of database guard are effected in the pfile that is created from spfile on primary.

A. Configuration details

The configuration of Oracle Data Guard consists of one primary database and up to thirty standby databases. The Oracle Data Guard members are connected by Oracle Net. The observer or broker provides a DGMGRL interface (Data Guard Manager Interface) and a graphical user interface.

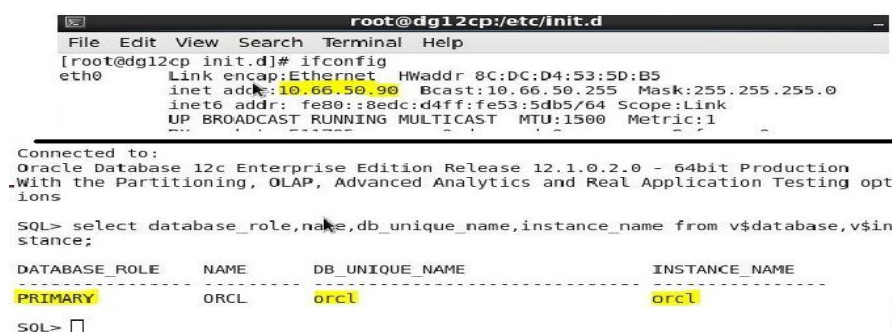
The Primary, Standby and Observer Systems are located at the following addresses

PRIMARY SERVER IP : 10.66.50.90

Since we have selected **create and configure database** option in our Primary server setup with database name **orcl** while installing 12C, the **db_unique_name** is also **orcl** by default.

STANDBY SERVER IP : 10.66.50.61

OBSERVER IP ADDRESS : 10.66.50.55



```

root@dg12cp:/etc/init.d
File Edit View Search Terminal Help
[root@dg12cp init.d]# ifconfig
eth0      Link encap:Ethernet  HWaddr 8C:DC:D4:53:5D:B5
          inet addr:10.66.50.90  Bcast:10.66.50.255  Mask:255.255.0
          inet6 addr: fe80::8edc:d4ff:fe53:5db5/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> select database_role,name,db_unique_name,instance_name from v$database,v$instance;

DATABASE_ROLE  NAME      DB_UNIQUE_NAME  INSTANCE_NAME
-----
PRIMARY        ORCL      orcl            orcl

SQL>

```

Figure 4. Primary database configuration details

B. High Availability Implementation on Observer System

Connect to the Primary Server through **dgmgml** and configure Fast Start Failover. To connect Primary you must have **tnsnames.ora** file with both Primary and Standby addresses. As the Observer is installed with only 12C software **tnsnames.ora** will not be available to we need to configure **tnsnames.ora** using **netmgr** or create **tnsnames.ora** file. After the connection has established from Observer to Primary system create Fast Start Failover configuration. The Observer system details are shown in the below figure.



```

DGMGRL> enable fast_start failover
Enabled.
DGMGRL> show configuration
Configuration - dg12c_fsf
Protection Mode: MaxAvailability
Members:
  orcl - Primary database
  std  - (*) Physical standby database

Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS (status updated 15 seconds ago)

DGMGRL> show fast_start failover
Fast-Start Failover: ENABLED
Threshold:          30 seconds
Target:             std
Observer:           dgbroker.drcl.in
Lag Limit:         30 seconds (not in use)
Shutdown Primary:  TRUE
Auto-reinstat:     TRUE
Observer Reconnect: (none)
Observer Override:  FALSE

Configurable Failover Conditions
Health Conditions:
  Corrupted Controlfile          YES
  Corrupted Dictionary           YES
  Inaccessible Logfile           NO
  Stuck Archiver                 NO
  Datafile Offline               YES

Oracle Error Conditions:
(none)

DGMGRL>

```

Figure 5. High Availability Details on Observer System

C. Keepalived for IP failover

Keepalived is used for IP failover between two servers. It facilitates load balancing and **high-availability** of Server based systems. It works on Virtual Redundancy Routing Protocol (VRRP). We have running two servers with Data Guard in it and all we need to do is to make the **IP switchover** to standby when the Primary is not

available. The configuration of Keepalived consists of one leader server and one or more backup servers. The below example shows how the virtual IP address 10.66.50.250 is initially assigned to the master server (10.66.50.90). When the master server fails, the backup server (10.66.50.61) becomes the new master server and is assigned the virtual IP address 10.66.50.250. On RHEL 6.5 as a base repository. Install it using yum command line tool.

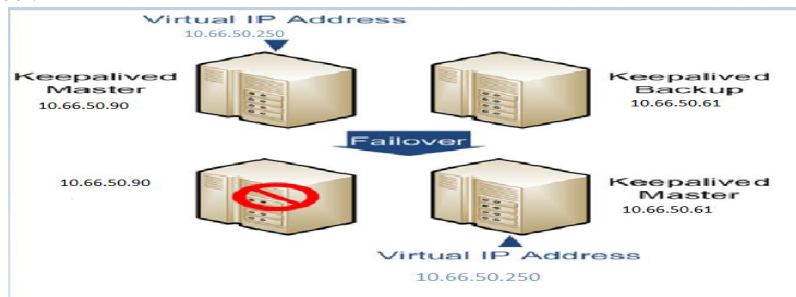


Figure 6. IP failover using Keepalived

Once this configuration is finished it will create a folder named keepalived in /etc/init.d location. The keepalived folder consists of configuration file named **keepalived.conf**. We need to edit that file in order to make IP failover.

In the event that the primary server (master) is not active, keepalived will assign the virtual IP address 10.66.50.250/24 to the eth0 interface on the standby server (backup), which becomes the primary server. To determine whether a server is acting as the master, you can use the **ipaddr show eth0** command to verify that virtual address is active or not, for example:

```
[root@dgl2cp keepalived]# service keepalived restart
Stopping keepalived: [ OK ]
Starting keepalived: [ OK ]
[root@dgl2cp keepalived]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 8c:dc:d4:53:5d:b5 brd ff:ff:ff:ff:ff:ff
    inet 10.66.50.90/24 brd 10.66.50.255 scope global eth0
    inet 10.66.50.250/24 scope global secondary eth0
    inet6 fe80::8cdc:d4ff:fe53:5db5/64 scope link
    valid_lft forever preferred_lft forever
```

Figure 7. Virtual IP assigned to Primary server

III. EXPERIMENT AND RESULT

I want to create a table called oracle_dg12c by connecting to user scott in Primary and will insert some rows into the Primary from the Sql developer interface and see that the data has come to Standby Server once it is committed.

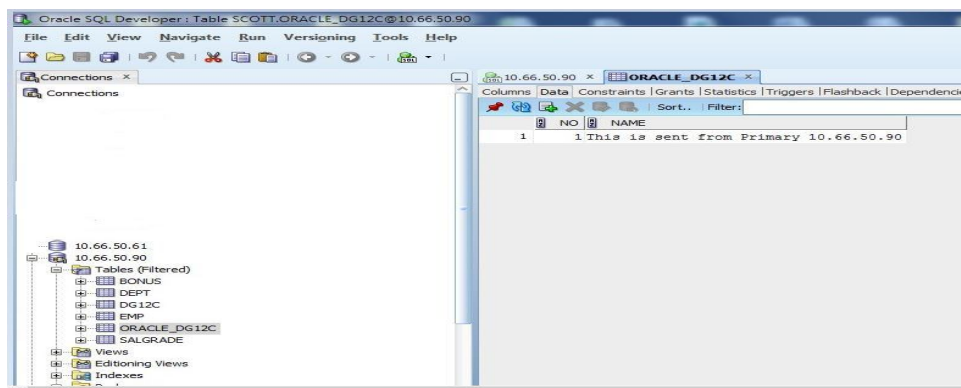


Figure 8. Sql Developer Interface connected to Primary Server

```

SQL> select database_role,protection_mode from v$database;
-----
DATABASE_ROLE      PROTECTION_MODE
-----
PRIMARY            MAXIMUM AVAILABILITY

SQL> select * from scott.oracle_dg12c;
-----
NO
NAME
-----
1
This is sent from Primary 10.66.50.90

```

Figure 9. Data committed into Primary Server.

```

SQL> select database_role,protection_mode from v$database;
-----
DATABASE_ROLE      PROTECTION_MODE
-----
PHYSICAL STANDBY  MAXIMUM AVAILABILITY

SQL> select * from scott.oracle_dg12c;
-----
NO
NAME
-----
1
This is sent from Primary 10.66.50.90

```

Figure 10. Data reflected to Standby Server Maintaing Consistency.

Row has successfully inserted into both Primary and Standby once the data that is inserted into Primary has committed.

Now testing Fast Start Failover, Before testing Fast Start Failover check the configuration of Primary in Observer System. Check whether the Fast Start Failover has enabled or not and most important is without starting the Observer Fast Start Failover won't happen. So start the Observer by issuing '**start observer**' command that is connected to Primary Server. Validate the database using '**validate database std**' and '**validate database orcl**' commands to ensure that the systems are ready to implement Failover and Switchover.

Since the observer or broker is connected to Primary system, Fast Start Failover is implemented only if the Primary Server is not available so for testing purpose am going to off the power for my Primary Server or issue shutdown abort command in sql prompt of Primary. Shutdown immediate cannot make Fast Start Failover.

Switchover or Failover occurred means the databases will shut down and start-up automatically so we have to reload listener because it points to the past database instance. If you import auto listener scripts there is no problem with this issue. Now am going to make my **Primary unavailable** by making the **services stop** or simply **off** my Primary without shutting down my Primary database. As per our system after 30 seconds Fast Start Failover will be implemented because by default **FastStartFailoverThreshold** property is 30. My virtual ip is assigned to Primary and as my Primary is not available, ip must failover to Standby database server as fast start failover is occurred, the associated screen is as follows.

```

[root@dg12cs ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 8c:dc:d4:53:5d:d9 brd ff:ff:ff:ff:ff:ff
    inet 10.66.50.61/24 brd 10.66.50.255 scope global eth0
    inet 10.66.50.250/24 scope global secondary eth0
    inet6 fe80::8edc:d4ff:fe53:5dd9/64 scope link
        valid_lft forever preferred_lft forever

```

Figure 11. Virtual ip assigned to Standby Server.



```

oracle@dgbroker:~
Ready for Switchover: Yes
Ready for Failover:  Yes (Primary Running)

Current Log File Groups Configuration:
Thread #  Online Redo Log Groups  Standby Redo Log Groups  Status
1          3 (orcl)                       3 (std)                   Insufficient SRLs

Future Log File Groups Configuration:
Thread #  Online Redo Log Groups  Standby Redo Log Groups  Status
1          3 (std)                   2 (orcl)                  Insufficient SRLs

DGMGRL> start observer
Observer started

12:23:18.84  Tuesday, April 05, 2016
Initiating Fast-Start Failover to database "std"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "std"
12:23:52.20  Tuesday, April 05, 2016

```

Figure 12. Broker Screen Connected to Primary

The above screen shows fast_start failover is implemented successfully without any manual intervention, as our Primary is not available the observer makes the available Standby as Primary since our new Primary is 'std'. Now we have only one system that is 'std' which is in Primary state, I want to add some data into 'std' and then am going to power on the former Primary that is 'orcl' in which it will be converted into Standby Server. Check the data and be sure that the added data on 'std' will be available on 'orcl' also.

```

login as: root
root@10.66.50.61's password:
Last login: Tue Apr  5 13:56:41 2016 from 10.66.50.51
[root@dgl2cs ~]# su - oracle
[oracle@dgl2cs ~]# sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Tue Apr 5 13:59:34 2016

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> select database_role from v$database;

DATABASE_ROLE
-----
PRIMARY

SQL> insert into scott.oracle_dgl2c values(3,'from old standby i.e new primary')
;
1 row created.
SQL> commit;
Commit complete.
SQL> select * from scott.oracle_dgl2c;

          NO
          NAME
-----
          2
This is from new Primary 10.66.50.61 after failover
          3
from old standby i.e new primary
          1
This is sent from Primary 10.66.50.90

SQL>

```

Figure 13. Level of Consistency

This data has to reflect on Standby database i.e old Primary ensuring the server maintain consistency. Now am going to power on the former Primary 'orcl'

```

oracle@dgbroker:~
File Edit View Search Terminal Tabs Help

oracle@dgbroker:~
1 3 3 Insufficient SRLs

Future Log File Groups Configuration:
Thread # Online Redo Log Groups Standby Redo Log Groups Status
1 3 (std) (orcl) 2 Insufficient SRLs

DGMGRL> start observer
Observer started

12:23:18.84 Tuesday, April 05, 2016
Initiating Fast-Start Failover to database "std"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "std"
12:23:52.20 Tuesday, April 05, 2016

14:10:54.82 Tuesday, April 05, 2016
Initiating reinstatement for database "orcl"...
Reinstating database "orcl", please wait...
Reinstatement of database "orcl" succeeded
14:11:50.77 Tuesday, April 05, 2016

```

Figure 14. Auto reinstatement of Primary

Since our former Primary has Reinstated to Standby successfully, now check whether the data has applied from new Primary to Standby Server 'orcl'.


```
[root@dg12cp init.d]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 8c:dc:d4:53:5d:b5 brd ff:ff:ff:ff:ff:ff
    inet 10.66.50.90/24 brd 10.66.50.255 scope global eth0
    inet 10.66.50.250/24 scope global secondary eth0
    inet6 fe80::8edc:d4ff:fe53:5db5/64 scope link
        valid_lft forever preferred_lft forever
```

Figure 15. IP Failover

Data is successfully restored on Standby, remember that the Observer System should always points to Primary Server only, after failover the Primary has changed from 'orcl' to 'std'. If you want to maintain new primary you can you use but stop the observer at old Primary that is 'orcl' and Start the observer at new Primary ('std'). If you want the old Primary that is new Standby 'orcl' to change the role to Primary Server than you can switchover from Standby to Primary from observer system using 'switchover to orcl'.

IV.CONCLUSION

By implementing Data Guard in Oracle that was configured on Redhat server we are making data availability across different servers when the actual server is not available. Hence the Maximum Availability Architecture is achieved by introducing dataguard in oracle along with keepalived which is used for automatic switching of Virtual IP base on Priority in Redhat Servers. Hence making Observer System connect to Primary Server Fast Start Failover is achieved when primary is not available. One of the Standbys will take the role of primary and lost primary is automatically transferred into Standby server by using Observer System and virtual IP is pointed to new Primary server determines Maximum Availability Architecture.

REFERENCE

- [1] Oracle Data Guard Concepts and Administration, 11g Release 2 (11.2) E41134-03 November 2014
- [2] Scott Jesse, Bill Burton, Bryan Vongray "Oracle Database 11g Release 2 High Availability", 2011, 978-0-07-175207-7.
- [3] Larry Carpenter, Joe Meeks, Charles Kim, Bill Burke, 2009, Oracle Data Guard 11g Handbook, 978-0-07-162148-9.
- [4] https://docs.oracle.com/cd/B19306_01/server.102/b14239/log_transport.htm#i1265666
- [5] Oracle® Data Guard Broker 11g Release2 (11.2) E40771-07 November 2015
- [6] Oracle® Data Guard Broker 12c Release 1 (12.1) E48241-08 November 2015.
- [7] Red Hat Enterprise Linux 7 Load Balancer Administration Co pyright © 20 15 Red Hat, Inc. and others.
- [8] Oracle MaximumAvailability Architecture (White Paper September 2011), Michael T. Smith http://www.google.ae/url?sa=t&rct=j&q=write%20operations%20with%20oracle%20active%20data%20guard&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fdatabase%2Ffeatures%2Favailability%2Fmaawp11gr1activedataguard1128199.pdf&ei=W6XuUPCdCOvs0gWMtIDwDg&usg=AFQjCNHwurnyrbQzD5xJf5_kD7HUqGvjhQ&bvm=bv.1357700187,d.d2k
- [9] Auf Akhtar, Syed S. Rizvi, and Laiali Almazaydeh, "Data Guard: A new approach for recovery and rolling upgrades", The International Journal of Database Management Systems (IJDBMS), February 2010 Vol.2, No.1.
- [10] Rab Nawaz, Tariq Rahim Soomro, "Role of Data Guard in High Availability Database Operations", International Journal of Applied Information Systems (IJASIS), Vol.5, April 2013.
- [11] Manpreet Kaur, Manjot Kaur, "Achieving Zero Failover Using Logical Standby Database in Oracle Data Guard", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol-4, Issue12 Dec 2014