Volume 4, Issue 12, December 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Achieving Zero Failover Using Logical Standby Database in Oracle Dataguard

Er. Manpreet Kaur*, Er. Manjot Kaur
CSE & CGC Gharuan, PTU Jalandhar
Punjab, India

Abstract— ***The Data Guard is monitoring tool which observes continuous activities of primary database with the help of observer. The data guard can be configured using observer with physical standby database or it can be configured using logical standby database. The physical standby zero failover is already achieved but objective of our research is to achieve zero failover using logical standby database by configuring broker/observer within it.***

Keywords— ***Switchover, Failover, Log shipping, Standby database, Primary database, Observer/broker, Redo files.***

I. INTRODUCTION

There are basically two methods to configure data guard, one is with the help of standby physical database and other is with the help of logical standby database. Until now, there has been only the physical standby database implementation, in which the standby database can be in either recovery mode or in read-only mode. A physical standby database is physically equivalent to the primary database, and, while the database is applying logs it cannot be opened for reporting and vice versa. A logical standby database has the same logical schema as the primary database but may have different physical objects, such as additional indexes.

A logical standby database is logically identical to the primary database. The logical standby database is updated using SQL statements. The tables in a logical standby database can be used simultaneously for recovery and for other tasks such as reporting, summations, and queries.

Location and Directory Structure of Primary and Standby Databases

One crucial aspect of the Data Guard environment is the number and configuration of the databases involved. Of particular importance are whether the primary and standby databases reside on the same computer system or on different systems The primary and standby systems have identical or different directory. The location of systems involved in the Data Guard environment has obvious implications for a disaster recovery strategy. For example, if the primary system in a data center is destroyed, then you cannot perform failover to a standby database unless it resides on a different system, which may or may not be in the same data center. In a worst case scenario, if the data center is completely destroyed, then you cannot perform a failover to a standby database unless the standby database is located on a different system in a remote location structures.

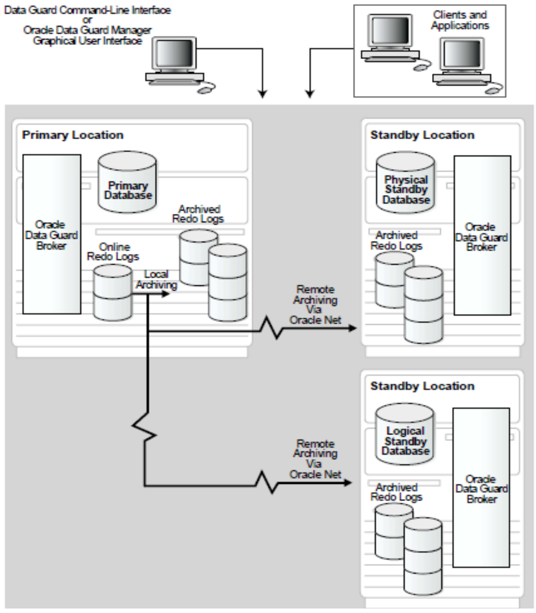


Fig 1 Architecture of Oracle Data Guard

To determine if the primary database contains data types or tables that are not supported by a logical standby database. Before setting up a logical standby database, you must make sure the logical standby database can maintain the data types and tables in your primary database.
The following lists indicate which of the various database objects are supported in logical standby databases.

Unsupported Datatypes
Logical standby databases do not support columns of the following data types:
NCLOB, LONG, LONG RAW, BFILE, ROWID, and UROWID, and user-defined type (including object types, REFs, v arrays, and nested tables). An error is returned if a logical standby database attempts to apply DML changes for a table that contains a column of an unsupported data type. To determine whether your primary database contains unsupported objects, use the
DBA_LOGSTDBY_UNSUPPORTED view. For example, enter the following SELECT statement on the primary database to list the names of primary database tables (and columns and data types in those tables) that are not supported by logical standby databases.
Ensure that table rows in the primary database can be uniquely identified. To maintain data in a logical standby database, SQL apply operations must be able to identify the columns that uniquely identify each row that has been updated in the primary database. Oracle Corporation recommends that you use primary keys in your primary database to ensure that SQL apply operations can efficiently and correctly apply data updates to a logical standby database. Most tables already have a primary key or a non-null unique index, and, if not, the supplemental logging feature (that you will enable in step 4) automatically gathers the information necessary to identify rows that have been updated in the primary database. With supplemental logging, information that will uniquely identify the row is added to every update transaction in the archived redo logs so that log apply services can properly maintain tables in the logical standby database.

II. RELATED WORK

Rab Nawaz[4] on his scrutiny gives the insight of high availability of data by using Data Guard. Rab and his companion Tariq Rahim shows how Data Guard is providing high availability with offloading workload to one or more standby databases and providing the opportunity to query real time data in standby databases. Related research was also done by Xiaochunn woo [5]. Woo with his companions show how Research and Implementation done on High-Availability Spatial Database Based on Oracle. The above researches show the maximum availability of database. Our research will show to achieve zero failover using logical standby database by configuring broker/observer within it.

METHODOLOGY OF CONFIGURING LOGICAL STANDBY DATABASE:

The supplemental information helps SQL apply operations to correctly maintain
your tables in the logical standby database. To verify whether supplemental logging
is enabled, start a SQL session and query the V$DATABASE fixed view. For example,
enter:

```
SQL> SELECT SUPPLEMENTAL_LOG_DATA_PK, SUPPLEMENTAL_LOG_DATA_UI FROM V$DATABASE;
SUP SUP
--- ---
YES YES
SQL>
```

View the SUPPLEMENTAL_LOG_DATA_PK, and SUPPLEMENTAL_LOG_DATA_UI
columns. If supplemental logging is not enabled, execute the following statements
on the primary database to add primary key and unique index information to the
archived redo logs, and to switch to a new redo log.

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX) COLUMNS;
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> ALTER SYSTEM SET LOG_PARALLELISM=1 SCOPE=BOTH;
```

Start the Resource Manager if you plan to perform a hot backup. If you plan to create the logical standby database using a hot backup copy of the primary database, you must start the Resource Manager. To do this, define the
RESOURCE_MANAGER_PLAN THE parameter to use a resource plan and then shut down and start up the primary database. If you do not have a resource plan, you can use one of the supplied plans.
Defining the SYSTEM_PLAN attribute. If the RESOURCE_MANAGER_PLAN the parameter was not defined when you started the instance, you must restart your primary database to ensure that the Resource Manager is running.
However, this can still be faster than performing a cold backup procedure.
The following example shows how to set the RESOURCE_MANAGER_PLAN
initialization parameter:

```
SQL>ALTERSYSTEMSET RESOURCE_MANAGER_PLAN=SYSTEM_PLAN SCOPE=BOTH;
SQL> SHUTDOWN
```

On the primary database, use the SQL CREATE TABLESPACE statement to create a new tablespace for the logical standby tables and use the DBMS_LOGMNR_D.SET_TABLESPACE procedure to move the tables into the new tablespace. The following example shows how to create a new tablespace named logmnrts$ and move the logical standby tables into that tablespace:

SQL> CREATE TABLESPACE logmnrts$ DATAFILE ’/usr/prod1/dbs/logmnrts.dbf’
2> SIZE 25 M AUTOEXTEND ON MAXSIZE UNLIMITED;
SQL>EXECUTE DBMS_LOGMNR_D.SET_TABLESPACE(’logmnrts$’);

Viewing logical activity

SQL apply operations for logical standby databases use a collection of parallel execution servers and background processes to perform a number of different tasks.

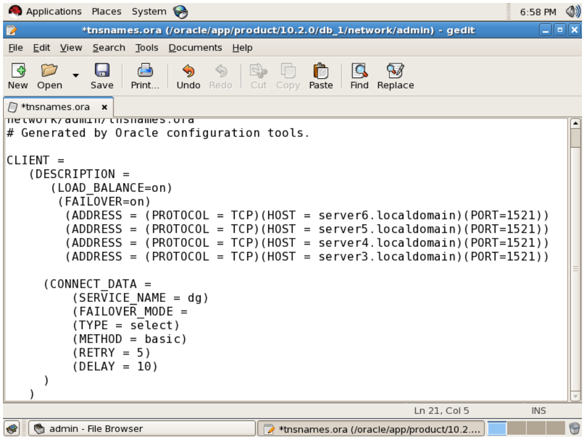
The V$LOGSTDBY view shows what each process is currently doing; the TYPE column describes the task being performed:

The READER process reads redo records from the archived redo logs.

The PREPARER processes do the heavy computing required to convert the block changes into table changes.

The BUILDER process assembles completed transactions.

Another place to gain information of current activity is the V$LOGSTDBY_STATS view, which provides state and status information. All of the options for the DBMS_LOGSTDBY.APPLY_SET procedure have default values, and those values (default or set) can be seen in the V$LOGSTDBY_STATS view.


Applications Places System
6:58 PM
*tnsnames.ora (/oracle/app/product/10.2.0/db_1/network/admin) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
*tnsnames.ora
Ln 21, Col 5
INS
admin - File Browser
*tnsnames.ora (/oracle/app/product/10.2....

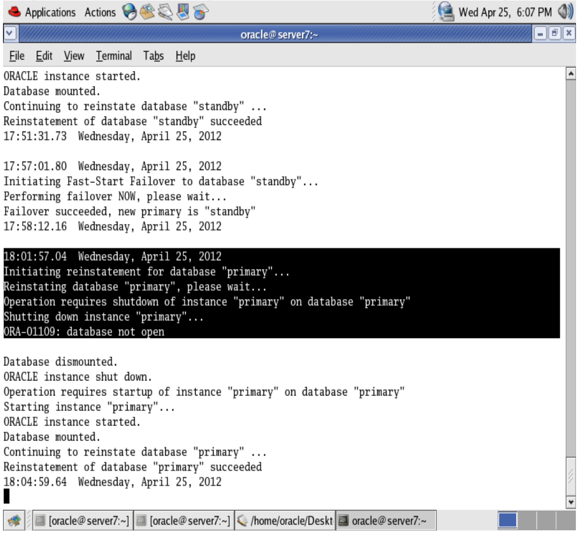
```
network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

CLIENT =
   (DESCRIPTION =
      (LOAD_BALANCE=on)
       (FAILOVER=on)
        (ADDRESS = (PROTOCOL = TCP)(HOST = server6.localdomain)(PORT=1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = server5.localdomain)(PORT=1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = server4.localdomain)(PORT=1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = server3.localdomain)(PORT=1521))

     (CONNECT_DATA =
         (SERVICE_NAME = dg)
         (FAILOVER_MODE =
         (TYPE = select)
         (METHOD = basic)
         (RETRY = 5)
         (DELAY = 10)
     )
   )
```

Fig 2 Changing parameters of TNS file for client connectivity

The DBA_LOGSTDBY_LOG view contains a list of all the log files that have been
registered. If you prefer using sequence numbers, rather than SCNs, Configuring and Managing Logical Standby Databases
DBA_LOGSTDBY_LOG view and the DBA_LOGSTDBY_PROGRESS view
to get that information. For example:
SELECT SEQUENCE# FROM DBA_LOGSTDBY_LOG l, DBA_LOGSTDBY_PROGRESS p
WHERE l.FIRST_CHANGE# <= p.APPLIED_SCN AND l.NEXT_CHANGE# > p.APPLIED_SCN;
SELECT SEQUENCE# FROM DBA_LOGSTDBY_LOG l, DBA_LOGSTDBY_PROGRESS p
WHERE l.FIRST_CHANGE# <= p.NEWEST_SCN AND l.NEXT_CHANGE# > p.NEWEST_SCN;
SELECT SEQUENCE# FROM DBA_LOGSTDBY_LOG l, DBA_LOGSTDBY_PROGRESS p
WHERE l.FIRST_CHANGE# <= p.READ_SCN AND l.NEXT_CHANGE# > p.READ_SCN;

III. FINAL RESULTS

To check the logical standby working we intentionally perform failover then we got the final result:



```
ORACLE instance started.
Database mounted.
Continuing to reinstate database "standby" ...
Reinstatement of database "standby" succeeded
17:51:31.73  Wednesday, April 25, 2012

17:57:01.80  Wednesday, April 25, 2012
Initiating Fast-Start Failover to database "standby"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "standby"
17:58:12.16  Wednesday, April 25, 2012

18:01:57.04  Wednesday, April 25, 2012
Initiating reinstatement for database "primary"...
Reinstating database "primary", please wait...
Operation requires shutdown of instance "primary" on database "primary"
Shutting down instance "primary"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "primary" on database "primary"
Starting instance "primary"...
ORACLE instance started.
Database mounted.
Continuing to reinstate database "primary" ...
Reinstatement of database "primary" succeeded
18:04:59.64  Wednesday, April 25, 2012
```

Fig. 4 Observer notify that Primary goes down and start initiating standby

After the failover we observer that client connectivity remain active, hence we achieve zero failover using logical standby database.

IV. DISCUSSION AND CONCLUSION

After watching performance of Oracle Dataguard with Observer we have experienced that with the help of Observer the management of data is very easy and to overcome any failure of primary data can be done by less human interaction. Data Guard is more efficient than any Data management tool exist till now because other tools require much more manual interaction. Before Dataguard the manual management of data is done, Then Recovery Manager and Restore options came to minimize the data loss. After that Oracle Dataguard proves to be a boon for Data Management but some manual work still requires in it. Then Observer improves the performance by monitoring the available standby and choosing standby appropriately and also decreases human interaction. And it is considered the best tool to utilize data and retrieve transactions but as technology is developing new tools each day, it is possible to enhance Dataguard Observer technique of switchover and Failover, Still after so many efforts it require human coding and modification of parameters by manager to function correctly. The further enhancement can be done by minimizing the connection establishment time. While synchronizing the Primary, Standby, Observer site is quite complex process. We need to create data services and start the listener then we can start the observer after that. So scripts can be used to reduce this work and Observer performance will be improved also. So we can expect in coming years a much improved and less complex version of Dataguard Observer which will minimize the human efforts and have stronger impact on Dataguard techniques.

ACKNOWLEDGMENT

I am bringing this report before you on “ACHIEVING ZERO FAILOVER USING LOGICAL STANDBY DATABASE IN ORACLE DATAGUARD ". Carrying out this project and its presentation in literary form becomes possible with the help, guidance and inspiration I received from the professionals engaged in education. I feel obliged to all those authorities whose work has been consulted and utilized and acknowledge the text. First of all I express my sincerest debt of gratitude to the **Almighty God** who always supports me in my endeavors. I would like to appreciate the pain staking efforts of **Er. Manjot Kaur** for educating and guiding me at each and every stage and providing me the information related to my chosen topic.

REFERENCES

[1] ORACLE, “Oracle9*i* Data Guard Concepts and Administration”, Release 2 (9.2), March 2002 Part No. A96653-01

[2] 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 (IJDMS), February 2010 Vol.2, No.1

[3] Ashish Ray, Ron Weiss, Tammy Bednar, Raymond Dutcher, Lawrence To, Juan Loaiza, “Oracle Dataguard disaster recovery for enterprise” March 2002, Oracle corporation, world headquarters, 500 oracle parkway, redwood shores, CA 94065. U.S.A

[4] Rab Nawaz, Tariq Rahim Somroo, “ Role of Oracle Active Data Guard in High Availability Database Operations”,.International Journal of Applied Information System, Foundation of Computer Science, New York, USA, 2013, Series 5, volume 5, 10.5120

[5] Xiaochun Wu, Kai Wang, Zixuan Su, Yanjun Liu Research and Implementation done on “High-Availability Spatial Database Based on Oracle”, IEEE

[6] Joe Meeks, Larry Carpenter, Ashish Ray, Oracle Dataguard with oracle database, October 2011, oracle corporation, world headquarters, 500 oracle parkway, Redwood shores, CA 94065. U.S.A

[7] Joseph Meeks, Michael T. Smith, Ashish Ray, Sadhana Kyathappala, Sean Connolly, “Oracle Data Guard 10g Release 2 Fast-Start Failover Best Practices” January 2010, Oracle Corporation,World Headquarters, 500 Oracle Parkway Redwood Shores, CA 94065 U.S.A.

[8] Hitachi-Oracle‟s BCM Platform Solution , “Verification Report on Oracle Active Data Guard”, March 2008 V: 1.0

[9] Oracle, “Oracle Database High Availability Best Practices 11*g* Release 2 (11.2)”, E10803-02.

[10] Emre Bransel, Nassyam Basha, “Oracle Data Guard 11gR2 Administratin Beginner‟s Guide” June 2013, ISBN: 1849687900 ISBN :139781849687904.

[11] Oracle, “Oracle Data Guard concepts and Administration 11g” release (B28294-03), August. 2008, pp. 25-288 [12]

[12] R. G. Liu, B. G. Pu, S. D. Du, S. N. Wang, "Research and application of high availability based on Mass spatial database", Computer Engineering, Vol. 34, March. 2008, pp. 263-265

[13] J. P. Chen, “Building Oracle HA environment framework, practice and experience summary of enterprise-Level HA database", Beijing: Electronic Industry Press, 2008, pp. 40

[14] Virginia Beecher, Viv Schupmann, Janet Stern, “Oracle Database High Availability Overview”, 11g Release 2 (11.2) 2011, E17157-08

[15] Larry M. Carpenter, “Oracle Open World Active Data Guard Hands on Lab”, October 2011,