

Troubleshooting Data Guard

In my last section on Data Guard we are going to cover troubleshooting, I cover common issues and resolutions regarding the configuration and maintainance of a Data Guard environment.

You already know that you have the [alert log and trace files](#), database health check which I covered in [monitoring](#), Data Guard [Broker](#) checks and configuration. Data Guard could have any of the following problems

- Common management issues
- Physical standby issues
- Logical Standby issues
- Switchover problems
- failover problems
- Data Guard Broker issues
- Data Guard snapshot standby issues

As I mentioned above I will tackle only the most common Data Guard problems, the internet should be able to help with most if not all problems, I am sure that someone out there has had the same problem and fixed it.

First we will start with log files, there are anumber of them to keep a check on

- Alert Log
- Observer Logs
- Data Guard trace files
- Data Guard Broker log files

To find the location of some of these logs use the command below

identify log files	col name for a25 col value for a65; select name, value from v\$diag_info where value like '%oracle%';
--------------------	---

If you are using the Fast-Start Failover (FSFO) you can check for additional logging in the observer log file, you can also check the configuration file called *FSFO.dat* they are normall kept in *\$ORACLE_HOME/dbs* and *\$ORACLE_HOME/rdbms/log* directories. when starting the observer you can state the logfile name, the file will be in the directory where you run the below command

observer logfile name	dgmgrl -logfile observer.log / "start observer"
-----------------------	---

Data Guard trace files can be turned on by the parameter *log_archive_trace*, you basically set the level that you want to trace at the higher the level the more detail, when set on the standby database it will capture output from RFS and MRP0 processes.

turn on Data Guard tracing	sql> alter system set log_archive_trace=16;
----------------------------	---

The Data Guard has a number of reports that you can run, these I covered in the [Broker](#) section.

Finally before we get in specific errors here is a list of views that might be useful

dba_logstdby_events	contains last 100 events that occurred on the logical standby
dba_logstdby_progress	check whether SQL Apply is progressing
dba_logstdby_log	checks whether archive logs are being delivered to a logical standby
dba_logstdby_unsupported	identifies SQL Apply unsupported data types
v\$archive_dest	Describes all the destinations in the Data Guard configuration
v\$archive_dest_status	Displays runtime and configuration information for the redo transport destinations
v\$archive_gap	Displays information to help you identify a gap in the archive redo log files on a physical database
v\$dataguard_config	Lists the <i>db_unique_name</i> parameters defined in the data guard configuration in <i>log_archive_config</i>
v\$dataguard_status	Displays and records events that would typically be triggered by any message to the alert log or server process trace files
v\$log	online redo log file information
v\$logfile	online redo log file and standby redo log file information
v\$log_history	contains archive log history information from the control file
v\$archive_log	contains more detail archived log information from the control file
v\$logstdby_progress	show whether logical standby pocess is running; if query returns no rows, its not running
v\$managed_standby	display current status information for Oracle database processes related to Data Guard
v\$standby_log	contains standby log file information

Configuration and Management Errors

We start with the password file error, if you see ORA-01034 or ORA16191 errors then check that the sys account password is set correctly and passwords are set the same across databases, if you are certain these are set correctly then try recreating the password file, also confirm that the parameter *remote_login_password* is set to either *shared* or *exclusive*.

if the SQL Apply fails with the error ORA-01031 tells you that SQL Apply has failed with insufficient privileges, you should see "grant sysdba to prod1_user" or something similar in the alert log.

granting privileges	<pre>sql> alter session disable guard; sql> grant sysdba to prod1_user; sql> alter session enable guard;</pre>
---------------------	---

There will be a time when you have to resolve a redo gap manually, normally this is automatic but in some cases it must be done manually. First you have to determine where the gap exists on your physical standby

determining the redo gap	<pre>## run this command on the physical standby select * from v\$archive_gap; thread# low_sequence# high_sequence# ----- 2 6233 6233 3 4531 4531 4 4938 4939</pre>
--------------------------	--

Using the example above we can see thread 2 is missing 6233 and thread 3 is missing 4531 and thread 4 is missing between 4938 and 4939, to identify the missing archive use the below command

identify missing archive logs	<pre>## run the below on the primary once you have the archive gaps that are missing select name from v\$archived_log where thread# = 2 and dest_id = 2 and sequence# = 6233; name ----- /u01/oradata/PROD1/recovery_area/arch_2_6233_656445787.arc</pre>
-------------------------------	--

Copy the missing archives from the primary to the standby and register them, then check that they have been applied

register the missing archive logs	<pre>## On the standby after you have copied the missing archives across from the primary sql> alter database resister logfile '<log file name>'; ## determine whether other archive log files are missing select max(r.sequence#) last_seq_recd, max(l.sequence#) last_seq_sent from v\$archived_log r, v\$log l where r.dest_id = 2 and l.archived='YES'; ## check that if gaps still exist select thread#, sequence#, file_name from dba_logstdby_log l where next_change# not in (select first_change# from dba_logstdby_log where l.thread# = thread#) order by thread#, sequence#;</pre>
-----------------------------------	---

Lets say that you have a standby database that has been out of action for a little while and you need to get it back up and running the steps are as follows

get a standby up and rolling again	<ol style="list-style-type: none"> stop the redo on the standby database sql> alter database recover managed standby database cancel; determine the current SCN for the standby database sql> select current_scn from v\$database; use the value for the current SCN to take your incremental backup on the primary rman> run { allocate channel d1 type disk; allocate channel d2 type disk; backup incremental from scn <scn number from above> database format '/tmp/dba/bkups/PROD1_%U'; release channel d1; release channel d2; } on the primary obtain a new standby control file and copy it to the physical standby sql> alter database create standby controlfile as '/tmp/stdby.ctl' copy the backup set and control from the primary to the standby use either scp or sftp, etc shutdown the standby and replace the controlfile and startup the standby in mount mode sql> startup mount catalog the backup using the target database controlfile rman> catalog start with '/tmp/dba/bkups'; recover the database from the incremental backup using noredo rman> recover database noredo; restart the MRP process sql> alter database recover managed standby database using current logfile disconnect;
------------------------------------	---

To check that the standby is receiving archive redo logs you run the following command

check that archive logs are being received	<pre>select dest_id, status, error from v\$archive_dest;</pre>
--	--

Check the following on the listener

- The listener.ora has been correctly configured
- The listener is running on the standby
- The service is not correctly configured in the tnsnames.ora file (check both the primary and standby)
- use tnsping and ping to check all connections

also check the archive destination for any problems

- the service name is specified in the *log_archive_dest_n* parameter for the primary database
- the *log_archive_dest_state_n* parameter for the standby is not set to *enable*
- a disconnect occurred and the the number of seconds specified by *reopen* has not yet passed
- make sure the standby is up and running in mount mode
- check that the standby database is not too out of sync otherwise use the above to get it rolling again

Physical Standby Issues

There are a number of common issues regarding a physical standby database

- You cannot mount the physical standby database
- The standby archive destination is not defined properly
- The standby site does not receive logs
- The standby site is not processing the logs

The physical standby cannot be mounted if the controlfile was created with an operating system created backup or backup created using an *alter database* statement without the *standby* options. it must be created with ones of the following *alter database create standby controlfile* statement or *rman backup current controlfile* with the *for standby* option.

The standby redo log files must be the same size as the primary, if the sizes are different and a logswitch occurs, then the physical standby will be shutdown if it is in maximum protection mode. The primary will become unsynchronized if it is in maximum availability mode you will receive the following error message in the alert log *no standby log files of size <#> blocks available*.

If you receive an ORA-16066 error message it is because the parameter *remote_archive_enable* is set to false, the parameter controls whether the archival of the redo logs to remote destinations is permitted. This parameter has been deprecated but make sure that it is set to true.

If you receive an ORA-16204 error message it means that the value in the parameter *log_archive_dest_n* is incorrect, double check this parameter.

Remote archival to standby database fails with ORA-01031, this is the result of a missing password file on the standby database, copy the password file from the primary to the standby database and restart the standby.

missing password file	sql> select status, error from v\$archive_dest;
-----------------------	---

If you receive an ORA-00326 error message, it means that media recovery has found an archive log which was generated after the required archive log, basically it needs the correct log and that the logs are out of order. MRP process will not allow this to happen but it can if you are applying manually. When you plan to manually recover a physical standby for any reason, you must first examine the standby control file to determine what archive logs have been registered for each thread. The you must make sure that you copy all archive logs from the primary that follow that last sequence for each thread, you **must not** rely on what you see on the disk - you must use the control file on the standby.

Logical Standby Database Failures

Several problems can occur

- SQL Apply stops
- SQL Apply hangs
- SQL Apply cannot keep up with the primary
- Data is not applied to the logical standby

Log apply cannot apply unsupported DML statements, DDL statements and Oracle-supplied packages to logical standby databases running SQL Apply. You can determine what has failed with the below statement

What has failed	<pre>select xidusn, xidslt, xidsqn, status, status_code from dba_logstdby_events where event_time = (select max(event_time) from dba_logstdby_events); ## Logical standby example of an error XIDUSN XIDSLT XIDSQN STATUS STATUS_CODE ----- 8 9 900 ORA-16226: DDL skipped due to lack of support 16226</pre>
-----------------	---

The above statement will display what cause the SQL Apply to stop, remember that you cannot randomly skip errors or transactions in SQL Apply without first fixing the issue that caused the problem in the first place, so never randomly skip transactions at a logical standby never.

ORA-01403 no data found errors is when DML is executed on the logical standby to tables maintained by SQL Apply, the primary and logical standby can get out of sync and the error won't show up until the table on the primary standby is updated. to resolve the issue you would need to skip and re-instantiate the table, if you have a database link created to connect to the primary, you can use the `dbms_logstdby.skip` and `dbms_logstdby.instantiate_table` procedure.

no data found	<pre>sql> exec dbms_logstdby.skip('DML','ACTORS','%'); sql> exec dbms_logstdby.instantiate_table ('PROD_USER', 'ACTORS', 'PROD1_DBLINK');</pre>
---------------	---

If it appears that SQL Apply is hanging for the below steps

SQL Apply hanging	<ol style="list-style-type: none"> 1. check <code>dba_logstdby_progress</code> to see if any activity is coming through <pre>sql> select applied_scn, applied_time, read_scn, read_time, newest_scn, newest_time from dba_logstdby_progress;</pre> 2. check the <code>high_scn</code> from the <code>v\$logstdby</code> view, the scn should change as the SQL Apply progresses <pre>sql> select type, high_scn, status from v\$logstdby;</pre> 3. check <code>v\$logstdby_stats</code> for information on the primary
-------------------	---

Switchover Issues

If the switchover to the standby from the primary fails you probably have sessions connected you need to use the *with session shutdown* option, you can check what sessions or jobs are running, but the above option will allow you to switchover

It is also possible that the standby never received the last redo from the primary, use the below query to determine when the last redo was transmitted, if the last was not transmitted just copy it from the primary and register it, when you restart the log services it will then be applied automatically.

last redo transmitted	<pre>select thread#, sequence#, archived, applied, status from v\$archived_log;</pre>
-----------------------	---

If you get ORA-01102 error message make sure that the `db_unique_name` has been set on both the primary and the standby, if have not set it then set it and restart the database.

If you active sessions when switching over it may fail with a ORA-01093 error then use the *with session shutdown* clause as part of the *alter database commit to switchover to physical standby*. Other processes that prevent switchover are the following

- Job queue scheduler process
- Advanced queue time manager
- Oracle Enterprise manager agent

You may end up with physical standbys if the switchover fails in the middle, in this case go to the original primary and run the following

two physical databases after switchover	<pre>## On the original primary sql> alter database commit to switchover to primary;</pre>
---	---

If you want to back out from an unsuccessful switchover on a physical standby then try the below

backed out switchover on physical standby	<ol style="list-style-type: none"> 1. Run the command below on the original primary, if this statement is successful stop and restart the database, if not successful then continue the next steps <pre>sql> alter database commit to switchover to primary;</pre> 2. When switchover was started a trace file was written in the log directory, this trace file contains the SQL statements needed to re-create the original primary control file. Capture the statements from this file and execute them from the SQL*Plus on the new standby database, the new standby database will revert back to the primary role. 3. Shut down the original physical standby and create a new standby control file on the primary database, copy the control file to the standby 4. Restart the original physical standby instance, hopefully the Fetch Archive Log (FAL) process will start and rearchive any missing archived redo log files to the physical standby.
---	---

After a switchover if the archive logs are not being applied that the following actions

archive redo logs are not applied after switchover	<ol style="list-style-type: none"> 1. verify that the <code>log_archive_dest_n</code> is set correctly <pre>sql> select dest_id, status, destination from v\$archive_dest;</pre>
--	--

- | | |
|--|---|
| | <ol style="list-style-type: none">2. verify the following are set correctly
<code>standby_archive_dest</code>
<code>log_archive_format</code>3. on the standby check the following
<code>db_file_name_convert</code>
<code>log_file_name_convert</code>
<code>standby_file_management</code>4. verify the <code>listener.ora</code> file is correct and the <code>tnsnames.ora</code> file is also correct5. verify that the listener is up and running on the standby |
|--|---|