

The Care and Feeding of a Standby Database

John Garmany

garmanyj@proxitec.com

Who Am I

John Garmany
Senior Consultant

ORACLE | CERTIFIED
PROFESSIONAL



- West Point Graduate – GO ARMY!
- Masters Degree Information Systems
- Graduate Certificate in Software Engineering

Standby Database

- An exact copy of the Primary Database.
 - Datafiles, tablespaces, users, grants, etc
- Updated along with the Primary.
 - Normally by the primary.
- May be local or remote or both.
 - Availability vs Disaster Recovery

Standby Database

- Availability Options
 - RAC
 - Replication
 - Cluster (Equipment Standby)
 - Standby
 - RMAN Recovery
- Fast and Expensive vs Slow and Cheap.

Availability Options

- 1993 Oracle 7 Introduced Manual Standby
 - Still used in Oracle7 thru 11g.
- 1996 Oracle Parallel Server
 - Became RAC
- 1996 Replication
 - Other masters continue after loss of a master.
- Oracle8i Managed Standby
 - Automated Standby
- Oracle9iR2 Streams Replication

Managed Standby Database

- Managed Standby in Enterprise Edition
- Manual Standby in Standard Edition
- Only RMAN in XE

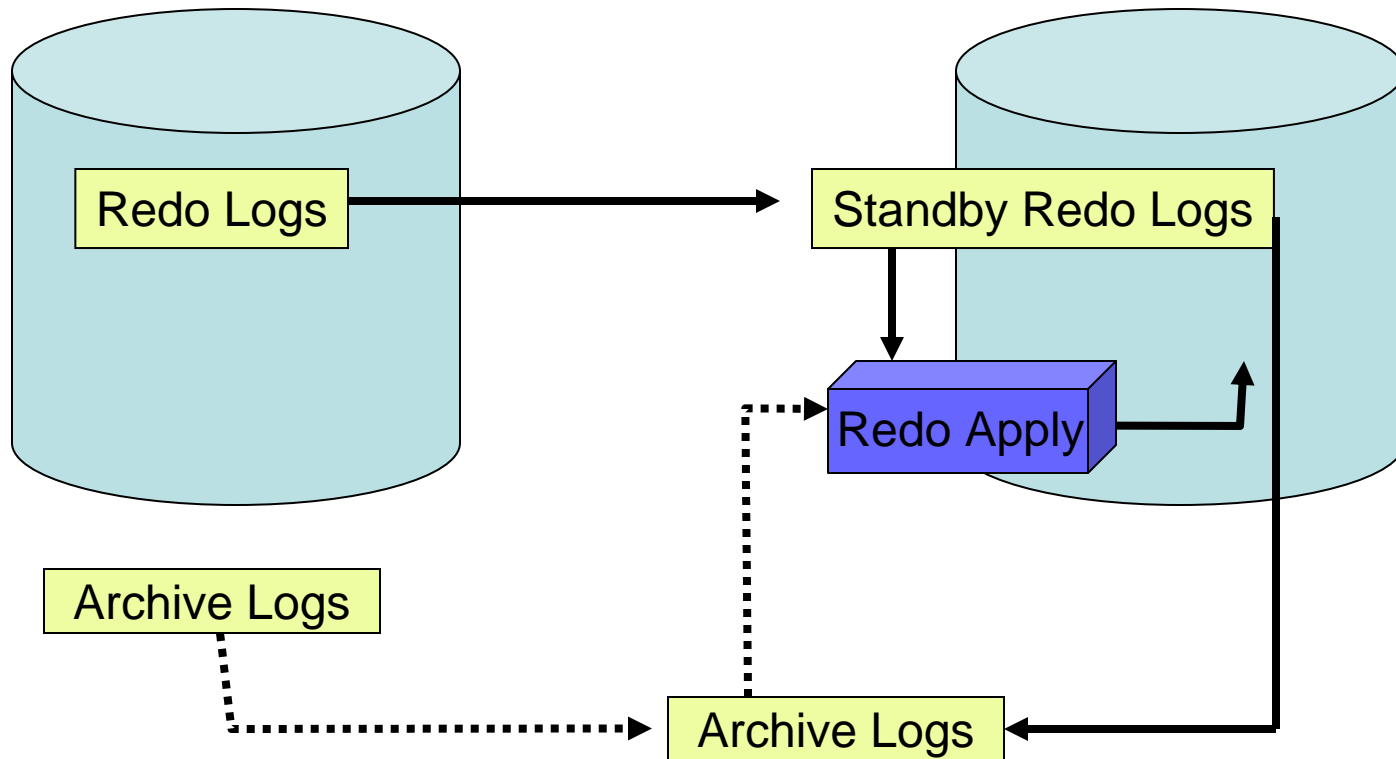
Standby Concepts

- Primary Database
 - Main Functional Database
 - Provides all database services
 - Licensed
 - Source Database

Standby Concepts

- Standby Database.
 - Target database or databases (up to 9)
 - Receives transactions from primary
 - Redo or Archivelogs
 - Transactions apply when in recovery mode.***
 - Can be opened in read-only mode.***
 - Must be licensed if used more than 10 days a year (may not be current).

Physical Standby



Standby Concepts

- Standby Database.
 - Used for Disaster Recovery
 - Can be used for Ad-hoc queries in read-only mode.
 - Can be used for RMAN backups
 - 10g+ Can be opened read-write and flashed back to before resetlogs.

Types of Standby Databases

- Physical Standby
 - Identical Structure
 - Update by recovery
 - Block by block match
- Logical Standby
 - All or part of the primary
 - SQL Apply
 - Structure does not match

Physical Standby

- Exact copy of the entire primary.
- Updated by Recovery of Archivelogs
- Manual
 - You must physically copy the archivelogs.
 - You must recover the standby
- Managed
 - Data Guard Does it all for you.

Physical Standby

- Read-only Mode
 - Open for ad-hoc queries
 - Recover stops while open
 - Recovery catches up when placed back in managed recovery mode.
- Open Read-write Mode
 - Open Resetlogs
 - Flashback to place back in Standby

Standby Lifecycle

1. Create Standby
2. Managed Recovery
 1. Open read-only
 2. Open read-write
 1. Flashback to Standby
3. Switchover to Primary
 1. Switchover to Standby
 2. Create new Standby
4. Failover to Primary
 1. Create new Standby

Switchover vs Failover

- Switchover
 - Primary and Standby available.
 - They switch roles and can switch back.
 - No loss of data.
- Failover
 - Loss of Primary
 - Activate Standby as Primary
 - May involve some data loss.
 - Primary discarded.

Data Guard Architecture

- Basic Steps
 1. User changes data.
 1. Changes written to online redo logs
 2. Archive writes changes to an archivelog
 2. Log Transport Process ships archivelog to Standby(s).
 1. Log Apply recovers the standby using the archivelog.
 2. If a log is missing, FAL server ask Log Transport to resend.

Data Guard Architecture

- Basic Steps with Standby Redo Logs
 1. User changes data.
 1. Changes written to online redo logs
 2. LWR (slave) write change to remote standby redo log.
 3. Standby create archivelog locally on log switch
 2. Log Transport Process ships archivelog to Standby(s).
 1. If a log is missing, FAL server ask Log Transport to resend.

The Log Writer

- Writes entries from the log buffer to the online redo logs.
- If Standby Redo Logs used, then it also write the entries to the standby redo logs
 - This is the slow part, remote write across the network.
 - Higher protection, slower commits.

The Remote File Server

- Runs on the Standby
- Receives the logs from the Log Transport Service.
- Receives redo from Log Writer if Standby Redo Logs are used.

Fetch Archive Log (FAL)

- FAL Client – Standby
 - Recover Process notifies FAL Client of missing Archivelogs.
 - FAL Client ask FAL Server to resent the log.
- FAL Server – Primary
 - Where to request Archivelogs
 - Shipped by Log Transport

Fetch Archive Log (FAL)

- Primary and Standby will have both FAL Server and FAL client defined. Only the appropriate process will be started.
- Support Switchover.
- If more than one standby, the primary is set to failover to only one. Must be reconfigured to failover to another.

Archive Gap Detection

- MUST apply Archivelogs in order.
- Can not jump an archivelog
- Loss of a log = recreate the standby.
- Missing log will stop Standby update.
 - Standby will wait for the log.
 - If the log is removed (deleted after backup, etc) it must be restored to recover the standby.

Network Requirements

- All communication between Primary and Standby is over Oracle Net.
- Primary load is transport of archivelogs.
 - Must transport them faster than database creates them.
 - Distance is not an issue.
- Standby Redo Logs can greatly increase network requirements.

Standby Administration

- Monitor, Monitor, Monitor.
 - If the Standby falls behind and an archivelog is deleted, you must recreate the Standby.



Funny, the standby was working last year.

Standby Administration

- Recover Configuration Parameters

- Foreground Process – don't!

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;
```

Never returns.

- Foreground with Timeout

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE TIMEOUT 5;
```

Times out if archivelog does not arrive in 5 minutes.

- Background Process

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT  
FROM SESSION;
```

Standby Administration

- Recover Configuration
 - Parallel Recovery

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
PARALLEL 5;
```

Used to catch up when lots of logs need to be applied.

- Stopping Recover

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
CANCEL;
```

Standby Administration

- Open Standby in Read Only
 - From Recovery Mode

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
      CANCEL;
```

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

```
-- return to recovery mode.
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
      DISCONNECT FROM SESSION;
```

Standby Administration

- Open Standby in Read Only
 - From Shutdown

```
SQL> STARTUP;
```

```
SQL> STARTUP NOMOUNT;
```

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

Standby Administration

- REAL TIME APPLY
 - Must have Standby Redo Logs
 - Applies redo as it arrives vs log switches
 - Keeps Standby as up to date as possible.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE;
```

Standby Administration

- Return Standby to Recovery Mode
 - From Open Read Only

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
DISCONNECT FROM SESSION;
```

11g Real-Time Query

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
SQL> ALTER DATABASE OPEN;  
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
DISCONNECT FROM SESSION;
```

Flashback with Standbys

Both the primary and standby can use Flashback Database.

If you flashback the primary, you must also flashback the standby. Not performed automatically.

Must be turned on before use.

Both the primary and standby have the same scns.

Flashback Primary

The Primary Database can be flashed back to previous scn. Requires open resetlogs.

Standby will follow the new instantiation path.

If the Standby has passed the point where the primary is flashed back to (normally happens) then the standby must also be flashed back to the same or earlier scn.

```
SQL> SHUTDOWN IMMEDIATE
```

```
SQL> STARTUP MOUNT EXCLUSIVE
```

```
SQL> FLASHBACK DATABASE TO TIMESTAMP(SYSDATE-(1/24));
```

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

Flashback with Standbys

A Physical Standby can be opened in read-write mode. It must be flashed back to a time before it was opened prior to placing it into Recover Mode. First place both databases in flashback mode.

```
SHUTDOWN IMMEDIATE
```

```
STARTUP MOUNT EXCLUSIVE
```

```
ALTER DATABASE FLASHBACK ON;
```

```
ALTER DATABASE OPEN; ALTER DATABASE RECOVER...
```

Flashback Standby

Create a flashback restore point. This is where we will flash back the database to place it back in recover mode.

```
SQL> create restore point before_open;
```

```
SQL> create restore point before_open guarantee  
flashback database;
```

You can also copy the current scn, date/time, etc

You can not use a restore point from the primary database on the standby database. Need to use scn or timestamp.

Flashback Standby

Copy the standby control file for use later. Opening the database will convert the standby controlfile to a normal controlfile.

Now open the database. We will cover this in detail later.

```
SQL> alter database recover managed standby database  
      finish force;  
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;  
SQL> alter database open;
```

Old Way - use above

```
SQL> alter database commit to switchover to primary;  
SQL> STARTUP MOUNT FORCE;
```

Flashback Standby

To return the database to standby mode:

Shutdown abort;

Startup mount;

Flashback the database.

Shutdown abort

** Replace the standby controlfile if needed

Convert Controlfile to Standby

Startup mount.

Start recovery.

Flashback Standby

```
SQL> shutdown abort;
```

```
SQL> startup mount;
```

```
SQL> flashback database to restore point before_open;
```

```
< SQL> shutdown abort;>
```

```
< replace controlfile with standby controlfile >
```

```
SQL> alter database convert to physical standby;
```

```
SQL> startup mount force;
```

```
SQL> alter database recover managed standby...
```

Standby Manual Recovery

- Manual Recovery of Standby Database.
 - From Shutdown

```
SQL> STARTUP NOMOUNT;
```

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

```
SQL> RECOVER STANDBY DATABASE;
```

Recover using archivelogs not in dest_1.

```
SQL> RECOVER FROM '/ORACLE/PROD/ARCH' STANDBY  
DATABASE;
```

Standby Administration

- Shutting Down the Standby Database.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
CANCEL;
```

```
SQL> SHUTDOWN IMMEDIATE;
```

To stop errors in the alert log on the Primary, defer the archivelogs before shutdown of Standby.

```
SQL> ALTER SYSTEM SET  
LOG_ARCHIVE_DEST_STATE_2=DEFER;
```

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Monitoring Physical Standby

- Alert Log on Primary and Standby
- OEM for Log Dest Status
- V\$ARCHIVED_LOG
 - Monitor on Standby not just Primary
 - No DBA view available on Standby since the database is not open.

Monitoring Physical Standby

- Recover Processes
 - v\$managed_standby

Select

```
PROCESS,  
SEQUENCE#,  
STATUS
```

From

```
V$MANAGED_STANDBY;
```

Monitoring Physical Standby

```
#!/bin/ksh
ORACLE_SID=db06
export ORACLE_SID
ORACLE_HOME=`cat /etc/oratab|grep $ORACLE_SID:|cut -f2 -d': '`
export ORACLE_HOME
ORACLE_BASE=/u01/app/oracle
export ORACLE_BASE

$ORACLE_HOME/bin/sqlplus /nolog<<!
connect / as sysdba
set pages 0 heading off feedback off
spool /tmp/standby_applied.lst
select max(sequence#) from v$archived_log where applied = 'YES';
spool off;
connect perfstat/perfstat@DB09.ORACLE.COM
set pages 0 heading off feedback off
spool /tmp/primary_current.lst
select max(sequence#) from v$archived_log;
spool off;
exit
!
```

Monitoring Physical Standby

```
applied=`cat /tmp/standby_applied.lst`
current=`cat /tmp/primary_current.lst`
statsvalue=$(( $current - $applied ))

echo $statsvalue
if [[ $statsvalue -gt 4 ]];
then
echo "The Standby Database is falling behind!" > /tmp/stdby_alert.lst
echo "Standby Applied Sequence: $applied" >> /tmp/stdby_alert.lst
echo "Primary Current Sequence: $current" >> /tmp/stdby_alert.lst
echo "Difference:                $statsvalue" >> /tmp/stdby_alert.lst
echo "" >> /tmp/stdby_alert.lst
df -h >> /tmp/stdby_alert.lst

echo "*****"
echo "The Standby is NOT Current"
# cat /tmp/stdby_alert.lst |mail -s "PROD Standby Monitor Alert *"
    prod_emergency@my_company.net
echo "*****"
fi
```

Tablespace and Datafile

- STANDBY_FILE_MANAGEMENT
 - Creates Datafile automatically.
 - If set to manual or there is a problem, DBA must assist.
 - Hot backup the datafiles to the standby.
 - Restart recovery.

Tablespace and Datafile

```
Primary SQL> ALTER TABLESPACE XXX OFFLINE;
```

Copy file to standby

```
Primary SQL. ALTER TABLESPACE XXX ONLINE;
```

```
Primary SQL> ALTER SYSTEM SWITCH LOGFILE;
```

```
Standby SQL> ALTER DATABASE RECOVER MANAGED STANDBY  
DATABASE DISCONNECT FROM SESSION;
```

Tablespace and Datafile

Deleting a Tablespace on the Primary.

```
Primary SQL> DROP TABLESPACE XXX;
```

```
Primary SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Wait until that archivelog (and maybe the next one) is applied. Then remove the files from the OS.

Forcing a Lag in the Apply

- Forcing a lag will allow you to stop the standby before a user error is applied.
- Lag is in the apply, not the delivery.

```
Standby SQL> ALTER SYSTEM SET  
LOG_ARCHIVE_DEST_2='SERVICE=stdby DELAY=180';
```

The OLD WAY. Use flashback instead!

Protection Mode

- Maximum Protection
 - No Data Loss
 - Redo Written to Online redo and at least one Standby redo log before commit returns
 - Primary Shuts down if unable to write to the Standby Log

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE  
PROTECTION;
```

Protection Mode

- Maximum Availability
 - No Data Loss (hopefully)
 - Redo Written to Online redo and at least one Standby redo log before commit returns
 - Primary reverts to Max Performance if fault on Standby log write.

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE  
AVAILABILITY;
```

Protection Mode

- Maximum Performance (the default)
 - No Data Loss (less hopefully)
 - Performance over total protection.
 - Commit returns after local redo write
 - Async write to Standby logs.

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE  
PERFORMANCE;
```

Role Transition

- Switchover
 - No Data Loss
- Failover
 - Possible Data Loss

Role Transition

- Preparation for Role Transition
 - Insure init.ora parameters are properly set.
 - Verify that the standby database is in archivelog mode.
 - Verify that TEMP tablespace exist on Standby
 - Remove any delay in applying logs.
 - Shutdown all but one RAC instance.
 - Determine if Standby is current.
 - V\$dataguard_stats

Switchover

- Two Phases
 - Primary to new Standby
 - Standby to new Primary
- Standby must be mounted before starting the switchover.

Switchover Primary

Verify Primary Switchover Status

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

TO STANDBY = ready for switchover.

SESSIONS ACTIVE = kick off the users.

Others indicate a problem that must be resolved before switchover.

Switchover Primary

Initiate the Switchover

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL  
STANDBY (WITH SESSION SHUTDOWN);
```

Converts the database to a standby database.

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP MOUNT;
```

- Now both are standbys.

Switchover Standby

Verify Primary Switchover Status

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

TO PRIMARY = ready for switchover.

SESSIONS ACTIVE = kick off the users.

Others indicate a problem that must be resolved before switchover.

Switchover Standby

Initiate the Switchover

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY  
      (WITH SESSION SHUTDOWN);
```

Converts the database to the Primary.

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP;
```

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Failover

- Primary Unavailable.
- May involve data loss.
- Normally write-off the old Primary.

Failover Standby

- Prep for Failover

Set Database to default protection.

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMUM  
PERFORMANCE;
```

Identify Log Gaps

```
SQL> SELECT THREAD#, LOW_SEQUENCE#, HIGH_SEQUENCE#  
FROM V$ARCHIVE_GAP;
```

If row returned, then resolve the gap.

Find the LOW#, HIGH# and all between
archivelogs.

Failover Standby

- Prep for Failover

Register the missing archive logs.

```
SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'XXXXX';
```

Repeat until no gaps.

```
SQL> SELECT UNIQUE THREAD# AS THREAD, MAX(SEQUENCE#) OVER  
      (PARTITION BY thread#) AS LAST FROM V$ARCHIVED_LOG;
```

Move any additional logs to the Standby and register them.

Failover Standby

- Prep for Failover

Complete the Recover.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
FINISH FORCE;
```

Terminates the recover when completed.

Convert the database to Primary

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
```

Failover Standby

- Open the new Primary.

If the database has not be opened read-only since the last startup:

```
SQL> ALTER DATABASE OPEN;
```

If the Standby was opened read-only since last startup:

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP;
```

Failover

- Once completed, make a backup!
- Old Primary:
 - Recreate as a new Standby Database.
 - Recover using a backup to a point prior to the Failover and convert to Standby.
 - Recover using Flashback Database to a point prior to the Failover and convert to Standby.

Backup with RMAN

- Physical Standby Only.
- Backup Standby, use to recover both Standby and Primary.
- Only Archivelogs created after Standby can be backup up on Standby.
- Must still backup Primary Control File if backup is taken of Standby.

Data Guard Broker

- Automates Standby Tasks
- Uses Oracle Enterprise Manager as GUI
 - The stand alone one
 - Not dbcontrol
 - Not the Java based OEM
- Command Line Interface is DGMGRL
 - This is what we use.
- Broker 10g+ is very different that 9i.

Contact Information

garmanyj@proxitec.com

www.proxitec.com

