Disaster Recovery with Oracle Data Guard

Gary Fox
Database Administrator – Team Lead
Xerox, Wilsonville

Disaster Recovery with Oracle Data Guard Agenda

- Overview
- Various Options
- Requirements
- Management and Maintenance
- Initial Setup
- Views

Who am I?

- Worked with Oracle databases since version 6
- Database Administrator since version 7
- DBA Team Lead at Xerox in Wilsonville the past five years
- Designed and taught various classes for companies, colleges and communities
  - Oracle
  - Unix
  - Math
  - Folk dancing

Me and Data Guard

- Currently on a project to change our Disaster Recovery from a 3rd party site to internal ones
- Data Guard is the method we are using for Oracle databases
- Changed first one 2 years ago with production database in Oregon and standby in New York
- There are many other parts to the picture
    - Windows application servers or VMs
    - Unix zones
    - SAN block replication
    - NAS file replication
    - Network

Oracle Data Guard Overview

- Oracle Data Guard provides the ability to create and maintain Standby databases at one or more sites
- These protect Oracle databases from database and server failures as well as site disasters
- Failover to one of the alternate sites can be set to happen automatically (fast-start failover) or manually if the primary database is not usable
- Updates to Primary are reapplied in Standby as they occur

Oracle Data Guard Overview

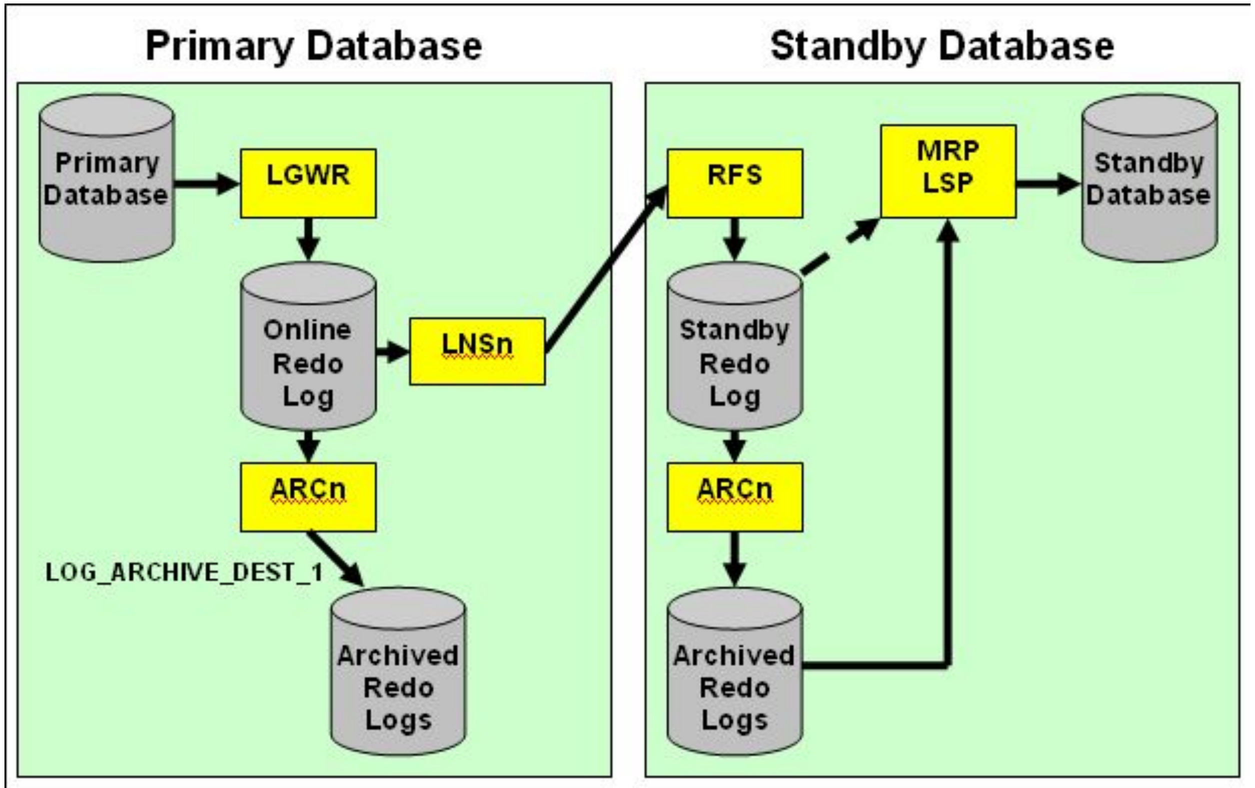
Advantages

- Recovery Point Objective (RPO) is very small or zero
- Recovery Time Objective (RTO) is measured in minutes
- Updates are done in Standby, so any potential physical corruption is not carried over
- If it gets behind, FAL client on Standby requests FAL server on Primary to send archive logs
  - FAL – Fetch Archive Log

Disadvantages

- Requires duplicate server, database, storage to be constantly in use
  - May be on the same server as the Primary
- Licenses for duplicates are needed
- People can't decide if data guard is one word or two

Oracle Data Guard



Background
Processes

LGWR	Log Writer
ARCn	Archiver
LNSn	Log Network Server (Data Guard specific)
RFS	Remote File Server (Data Guard specific)
MRP	Managed Recovery Process (Data Guard specific for Redo Apply)
LSR	Logical Standby Process (Data Guard specific for SQL Apply)

Standby Database Types

- Physical Standby – Redo Apply
  - An exact replica of Primary
    - Recovery applies changes block-for-block using the physical rowid
  - Is always running in recovery mode
  - Can be opened read-only by temporarily stopping Redo Apply
  - Redo Apply can be active while opened read-only with *Oracle Active Data Guard 11g*
    - Can be used for offloading read-only work from the primary database.
    - Requires extra license
  - Most examples in this presentation are from a physical standby

Standby Database Types

Logical Standby – SQL Apply

- Executes SQL statements to apply redo log data
  - Structure, indexes may be different
- Is open read-only
- Can be used for reports, backups, as well as disaster recovery
- The following are not supported
  - BFILE
  - Collections (including VARRAYS and nested tables)
  - Multimedia data types (including Spatial, Image, and Oracle Text)
  - ROWID, UROWID
  - User-defined types
  - eBusiness Suite
  - etc

Standby Database Types

Snapshot Standby

- Updateable
- Not in recovery mode
- Redo from Primary is not applied until Snapshot is converted to Physical Standby
  - Any updates made in Snapshot are discarded
- Enough space is needed for all unapplied logs

Oracle Data Guard Replication

- Sends transactions to one or more Standby databases
  - Can send transactions from redo logs as they are written (real-time apply)
  - Immediately applied on Standby
- If transactions fall behind, will automatically revert to log shipping replication
  - Massive updates on Primary database
  - OS patching of Standby server
  - Network problems
- Enterprise Manager can send alerts if gets more than a specified time behind

Oracle Data Guard Replication

Can be set to wait a specific time to protect against errors

```
log_archive_dest_2   = "SERVICE=mkslsb DELAY=60 "
```

Can be set without real-time apply

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

Can be set to use real-time apply

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE
  USING CURRENT LOGFILE DISCONNECT FROM SESSION;

ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE
   DISCONNECT FROM SESSION;
```

Protection Modes

Maximum Performance

Primary does not wait for an acknowledgement from Standby

Maximum Protection

Does not commit until redo written to at least one Standby

Creates guaranteed RPO of 0 seconds – no data loss

Causes potential slowness on Primary database as it has to wait for an acknowledgement from Standby

- Should have at least 2 Standbys

Maximum Availability

Is Maximum Protection unless a Standby fails

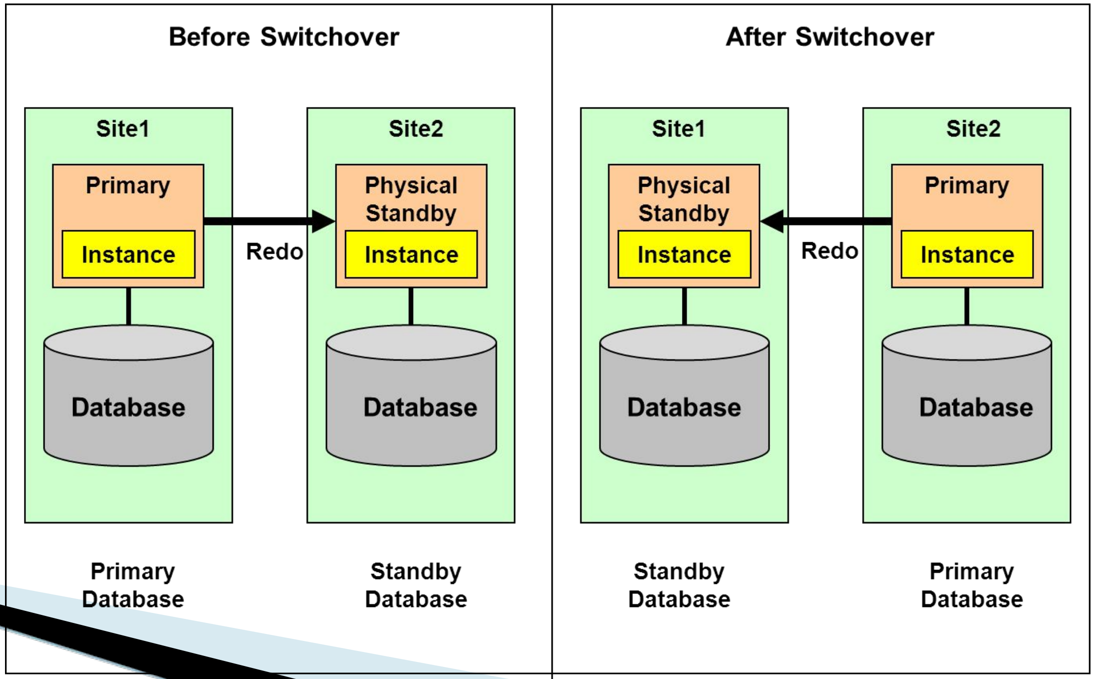
Then becomes Maximum Performance

Set in LOG_ARCHIVE_DEST_n

ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE {AVAILABILITY | PERFORMANCE | PROTECTION};

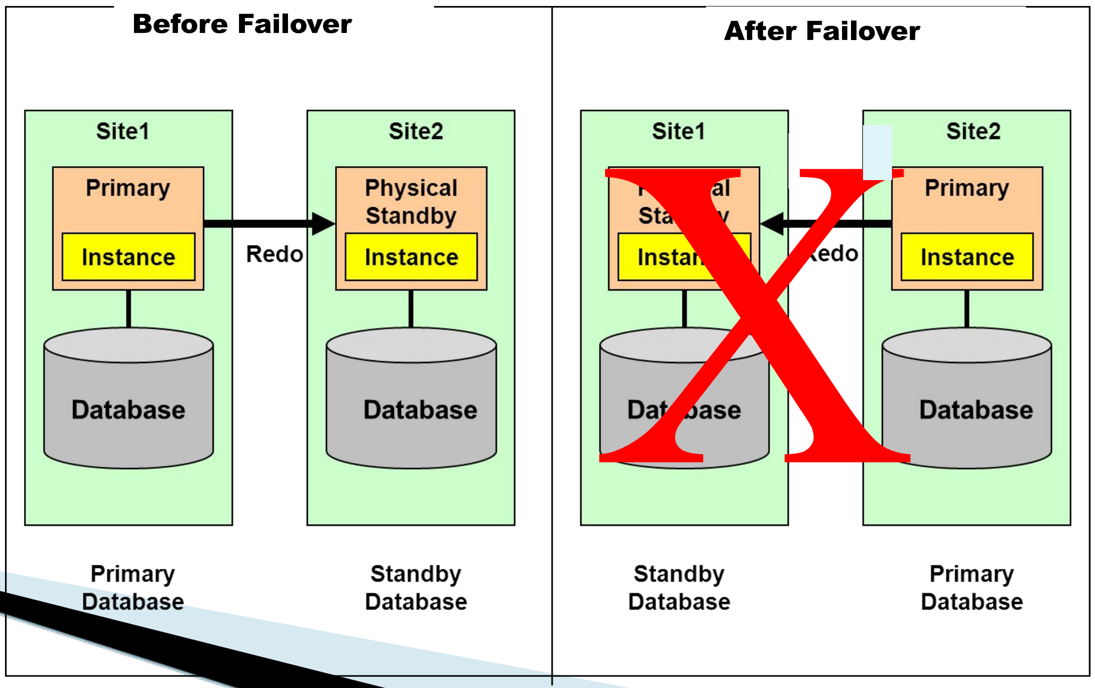
Switchover- planned

- This would be used for any testing, and for any disaster where a few minutes exist before shutdown
- No data loss
- Primary and Standby switch roles
  - Replication begins in reverse
- Can be gracefully “failed back” at will



Failover - unplanned

- Minimal expected data loss
  - Amount depends on network availability
- When the original source is rebuilt, it will take a full DB copy to the newly rebuilt server before reverse replication can begin



Failover - steps

Stop Redo Apply on the Standby database

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

Start failover and apply all received redo data

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

** Standby is now the Primary database

Verify that the Standby is ready to become a Primary

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

TO PRIMARY or SESSIONS ACTIVE indicates that the Standby database is ready to be switched to the Primary role

Requirements

- Hardware
  - Can be different between the primary and standby systems
  - Number of CPUs, memory size, storage configuration, etc
- Operating system must be similar
  - Some OSs can be different
  - OS release does not need to be the same
  - Either or both servers can be virtual or not
  - The directory structure does not need to be the same
- Databases
  - The databases must be in archivelog mode
  - They need to be Oracle Database Enterprise Edition
  - Both must be the same version, except during an upgrade
  - Either or both can be single instance or RAC

Server Requirements

These and some other heterogeneous platforms can be used together.

ID	PLATFORM_NAME Release name	PLATFORM_IDs supported when using Data Guard Redo Apply (Physical Standby)
2	Solaris OE (64-bit) Solaris (SPARC) (64-bit)	2
7	MS Windows (32-bit) MS Windows (x86)	7 8, 12 - Oracle 10g onward 10, 11, 13 - Oracle 11g onward, requires patch
8	MS Windows IA (64-bit) MS Windows (64-bit Itanium)	7, 12 - Oracle 10g onward 8 11, 13 - Oracle 11g onward, requires patch
10	Linux (32-bit) Linux x86	7 - Oracle 11g onward, requires patch 10 11, 13 - Oracle 10g onward
12	MS Windows 64-bit for AMD MS Windows (x86-64)	7, 8 - Oracle 10g onward 12 11, 13 - Oracle 11g onward, requires patch
13	Linux 64-bit for AMD Linux x86-64	7, 8, 12 - Oracle 11g onward, requires patch 10, 11, 20 - Oracle 10g onward 13

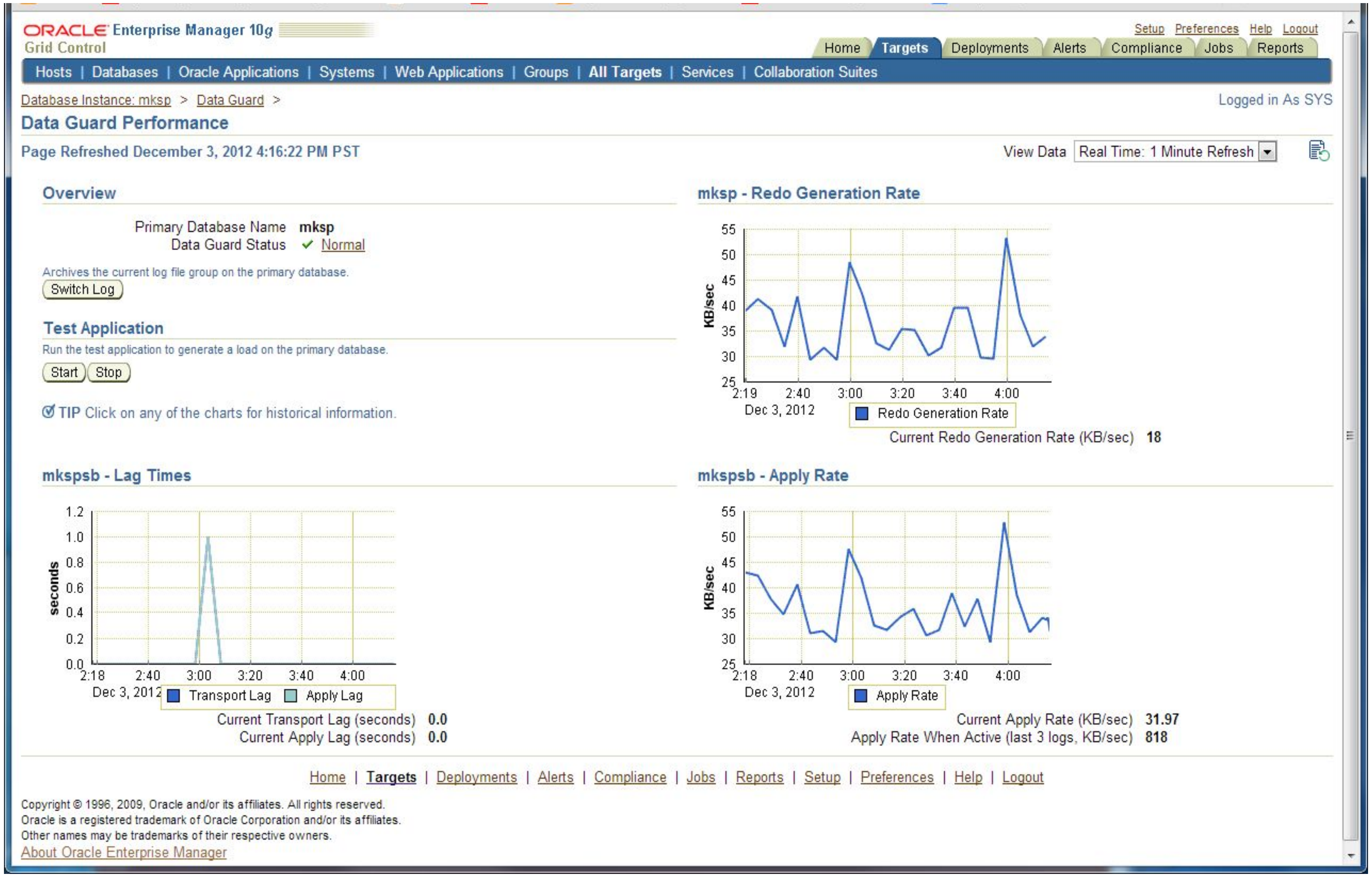
What Gets Replicated?

- Anything done in Primary database gets replicated to Standby
  - Adding or dropping tables, datafiles, users, etc
- Unix changes are not replicated
  - New mounts, init.ora, tsnnames.ora, application files
- Patching or upgrading databases
  - Patch/upgrade Standby binaries and restart Standby
  - Patch/upgrade Primary binaries
  - Run SQL patch/upgrade scripts in Primary
    - Normal Redo Apply runs them in Standby

Management and Maintenance

- Oracle Enterprise Manager
  - GUI interface
- SQL*Plus
  - Command line interface
- Data Guard Broker
  - Command line interface

Oracle Enterprise Manager



ORACLE Enterprise Manager 10g
Grid Control

Setup Preferences Help Logout

Home | Targets | Deployments | Alerts | Compliance | Jobs | Reports

Hosts | Databases | Oracle Applications | Systems | Web Applications | Groups | All Targets | Services | Collaboration Suites

Database Instance: mksp > Data Guard >

Logged in As SYS

Data Guard Performance

Page Refreshed December 3, 2012 4:16:22 PM PST

View Data: Real Time: 1 Minute Refresh

Overview

Primary Database Name	mksp
Data Guard Status	✓ Normal

Archives the current log file group on the primary database.

Switch Log

Test Application

Run the test application to generate a load on the primary database.

Start | Stop

TIP Click on any of the charts for historical information.

mksp - Redo Generation Rate

Current Redo Generation Rate (KB/sec) 18

mkspsb - Lag Times

Current Transport Lag (seconds) 0.0
Current Apply Lag (seconds) 0.0

mkspsb - Apply Rate

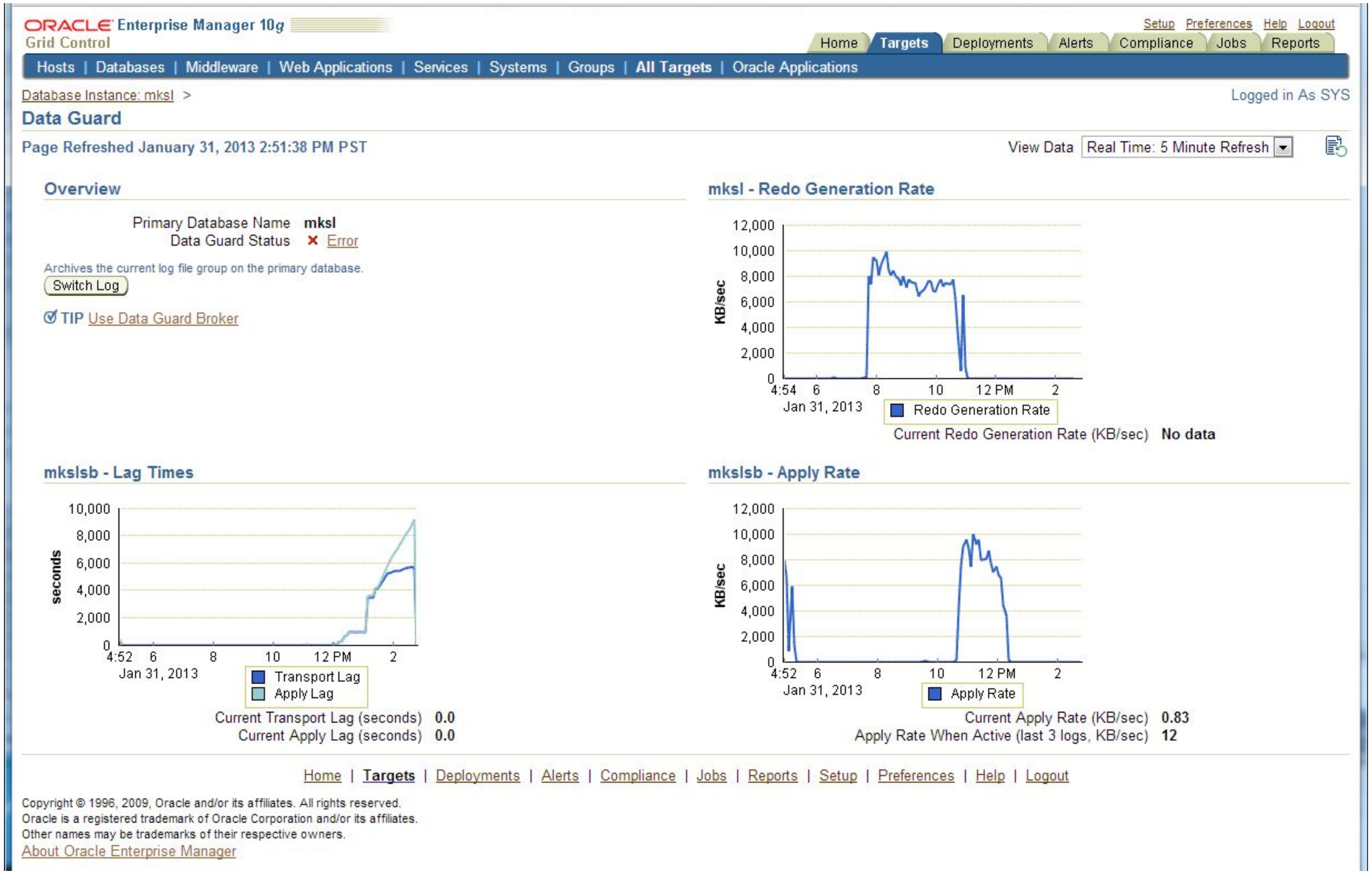
Current Apply Rate (KB/sec) 31.97
Apply Rate When Active (last 3 logs, KB/sec) 818

Home | Targets | Deployments | Alerts | Compliance | Jobs | Reports | Setup | Preferences | Help | Logout

About Oracle Enterprise Manager

The top is the Primary, the bottom is the Standby

Oracle Enterprise Manager


ORACLE Enterprise Manager 10g
Grid Control
Setup | Preferences | Help | Logout
Home | Targets | Deployments | Alerts | Compliance | Jobs | Reports
Hosts | Databases | Middleware | Web Applications | Services | Systems | Groups | All Targets | Oracle Applications
Database Instance: mksl >
Logged in As SYS
View Data: Real Time: 5 Minute Refresh
Switch Log
mksl - Redo Generation Rate
KB/sec
Redo Generation Rate
mkslsb - Lag Times
seconds
Transport Lag
Apply Lag
mkslsb - Apply Rate
Apply Rate
Jan 31, 2013

Data Guard

Page Refreshed January 31, 2013 2:51:38 PM PST

Overview

Primary Database Name	mksl
Data Guard Status	✕ Error

Archives the current log file group on the primary database.

TIP Use Data Guard Broker

Current Redo Generation Rate (KB/sec)	No data
Current Transport Lag (seconds)	0.0
Current Apply Lag (seconds)	0.0
Current Apply Rate (KB/sec)	0.83
Apply Rate When Active (last 3 logs, KB/sec)	12

Home | Targets | Deployments | Alerts | Compliance | Jobs | Reports | Setup | Preferences | Help | Logout

During this time the standby stopped applying logs as the disk was full

Oracle Enterprise Manager - Alerts

These are some of the alerts that EM will send

Target Name=mksl

Message=The Data Guard status of mksl is Error ORA-16778: redo transport error for one or more databases.
Metric=Data Guard Status

Target Name=mksl
Message=The Data Guard status of mksl is Error ORA-16662: network timeout when contacting a database.

Target Name=mksl

Message=The Data Guard status of mkslsb is Error ORA-16198: Timeout incurred on internal channel during remote archival.

Initial Standby Creation

- Initial Standby database needs to be created as a copy of the Primary
  - Can use rman duplicate
  - For very large databases or slow network can get backup to DVD/tape and overnight to remote site
- Standby needs its own Control File
  - ALTER DATABASE CREATE STANDBY CONTROLFILE AS '..';
- Create Standby listener
  - If using Data Guard Broker edit listeners
    - Primary (GLOBAL_DBNAME=<primary>_DGMGRL)
    - Standby (GLOBAL_DBNAME=<standby>_DGMGRL)
- Startup mount and start recovery
  - ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT FROM SESSION;

Initialization Parameters

A few of the parameters used in the Standby

```
db_name                    = <name of Primary>
db_unique_name             = <name of Standby>
fal_server                 = <name of Primary>
standby_file_management    = { auto | manual }
```

Auto allows datafiles to be automatically added on Standby when done on Primary

```
db_file_name_convert      =
   '/mksl/oradata/veksldata/','/u02/mkslsb/oradata/mkslsbdata/'
log_file_name_convert     =
   '/mksl/oradata/veksldata/','/u02/mkslsb/oradata/mkslsbdata/’
```

```
log_archive_min_succeed_dest= 1
log_archive_config        = "DG_CONFIG=(mksl, mkslsb)"
```

Set DG_CONFIG to a text string that contains the comma separated DB_UNIQUE_NAME of each database in the configuration

Initialization Parameters

log_archive_dest_n has many options

LOCATION – local,   SERVICE- remote

DELAY

AFFIRM SYNC (Max Protection)

NOAFFIRM ASYNC (Max Performance)

MAX_CONNECTIONS

VALID_FOR = (ONLINE_LOGFILE | STANDBY_LOGFILE | ALL_LOGFILES,
                    PRIMARY_ROLE | STANDBY_ROLE | ALL_ROLES)

```
log_archive_dest_1   = "LOCATION=/u02/mkslsb/lg01/mkslsbarch/
       VALID_FOR=(ALL_LOGFILES,  ALL_ROLES)
       DB_UNIQUE_NAME=mkslsb "
log_archive_dest_2   =   "SERVICE=mksl         ASYNC
       VALID_FOR=(ONLINE_LOGFILE,   PRIMARY_ROLE)
       DB_UNIQUE_NAME=mksl "
log_archive_dest_state_1   = enable
log_archive_dest_state_2   = enable
```

Data Dictionary Views

- There are many data dictionary views that are relevant to data guard

- The next few slides show some of them.
  - The SQL used is in the notes.

V$DATABASE

On Primary database

```
NAME       OPEN_MODE    DATABASE_ROLE        DB_UNIQUE_NAME
---------- ------------ -------------------- ---------------
MKSL       READ WRITE   PRIMARY              mksl
```

On Standby database

```
NAME       OPEN_MODE    DATABASE_ROLE        DB_UNIQUE_NAME
---------- ------------ -------------------- ---------------
MKSL       MOUNTED      PHYSICAL STANDBY     mkslsb
```

V$ARCHIVE_DEST

```
DEST_NAME            STATUS       DESTINATION                  TARGET
-------------------- ------------ ---------------------------- --------
ARCHIVER  SCHEDULE REOPEN_SECS DELAY_MINS MAX_CONNECTIONS VALID_TYPE
--------- -------- ----------- ---------- --------------- ---------------
VALID_ROLE     DB_UNIQUE_NAME
-------------- ---------------
LOG_ARCHIVE_DEST_1   VALID        /u02/mkslsb/lg01/mkslsbarch/  LOCAL
ARCH      ACTIVE           300          0               1 ALL_LOGFILES
ALL_ROLES      mkslsb

LOG_ARCHIVE_DEST_2   VALID        mksl                          REMOTE
LGWR      PENDING          300          0               1 ONLINE_LOGFILE
PRIMARY_ROLE   mksl

STANDBY_ARCHIVE_DEST VALID        /u02/mkslsb/lg01/mkslsbarch/  LOCAL
ARCH      ACTIVE           300          0               1 ALL_LOGFILES
ALL_ROLES      NONE
```

V$STANDBY_LOG

```
THREAD#   GROUP#  SEQUENCE# STATUS     ARCHIVED     FIRST_CHANGE#
  FIRST_TIME                        NEXT_CHANGE# LAST_TIME
  ------- -------- ---------- ---------- --------- ----------------
  -------------- -------------------- --------------
      1        4       4520 ACTIVE     YES           10907249776636
  31-JAN 10:01:52        10907249986689 31-Jan 10:58:14
      1        5          0 UNASSIGNED NO
      1        6          0 UNASSIGNED NO
      1        7          0 UNASSIGNED NO
```

V$ARCHIVE_GAP

```
THREAD#      LOW_SEQUENCE# HIGH_SEQUENCE#
---------- ------------- --------------
         1          3048           3050
```

V$DATAGUARD STATS

```
NAME                        VALUE              DATUM_TIME
------------------------ ---------------- ------------------------
apply finish time           +00 00:02:00.605
apply lag                   +00 01:47:30      01/31/2013 10:58:18
estimated startup time      14
transport lag               +00 01:29:36      01/31/2013 10:58:18
```

APPLY FINISH TIME - An estimate of the time needed to apply all received, but unapplied redo from the primary database.

APPLY LAG – How long the data in a standby database lags behind the data in the primary database.

ESTIMATED STARTUP TIME - An estimate of the time needed to start and open the database in seconds.

TRANSPORT LAG – A measure of the transport of redo to the standby database lags behind the generation of redo on the primary database.

V$ARCHIVED_LOG

```
REG CREA THRD APLD  SEQ#   FIRST_CHANGE#    NEXT_CHANGE# COMPT_TIME
--- ---- ---- ---- ----- --------------- --------------- ---------------
RFS ARCH    1 YES    782  10906917578148  10906920384710 30-JAN 17:01:09
. . . . .
RFS ARCH    1 YES   3607  10907223999049  10907223999089 31-JAN 09:01:36
RFS ARCH    1 NO    3608  10907223999089  10907223999134 31-JAN 09:01:52
RFS ARCH    1 YES   3608  10907223999089  10907223999134 31-JAN 09:01:42
RFS ARCH    1 YES   3609  10907223999134  10907223999171 31-JAN 10:01:37
RFS ARCH    1 YES   3610  10907223999171  10907223999216 31-JAN 10:01:44
```

APPLIED: IN-MEMORY, log file has been applied in memory, but datafiles have not yet been updated.

APPLIED: YES, datafiles have been updated.

V$ARCHIVE_DEST_STATUS

```
DEST_NAME              RECOVERY_MODE STATUS     ARCHIVED_THREAD#
  ARCHIVED_SEQ# APPLIED_THREAD# APPLIED_SEQ#
-------------------- ------------- ---------- ----------------
  ---------- --------------- ------------
LOG_ARCHIVE_DEST_1   IDLE          VALID                     1
        4519               0            0
LOG_ARCHIVE_DEST_2   IDLE          VALID                     0
           0               0            0
STANDBY_ARCHIVE_DEST IDLE          VALID                     1
        3773               1         3608
```

V$DATAGUARD STATUS

```
TIME            ERROR_CODE MESSAGE
--------------- ---------- ---------------------------------------
22:59:22 03-FEB          0 ARC0: Archival started
22:59:23 03-FEB          0 ARC1: Archival started
22:59:23 03-FEB          0 ARC2: Archival started
22:59:23 03-FEB          0 ARC1: Becoming the 'no FAL' ARCH
22:59:23 03-FEB          0 ARC2: Becoming the heartbeat ARCH
22:59:23 03-FEB          0 ARC2: Becoming the active heartbeat ARCH
22:59:24 03-FEB          0 ARC3: Archival started
22:59:29 03-FEB          0 Attempt to start background Managed Standby
  Recovery process
22:59:29 03-FEB          0 MRP0: Background Managed Standby Recovery
  process started
22:59:31 03-FEB          0 Primary database is in MAXIMUM PERFORMANCE
  mode
22:59:31 03-FEB          0 RFS[1]: Assigned to RFS process 5396
22:59:31 03-FEB          0 RFS[2]: Assigned to RFS process 5398
```

V$DATAGUARD STATUS

```
22:59:34 03-FEB          0 Managed Standby Recovery starting Real Time
  Apply
22:59:34 03-FEB          0 ARC3: Beginning to archive thread 1
  sequence 4545 (10908513865971-10908582304950)
22:59:35 03-FEB          0 ARC3: Completed archiving thread 1 sequence
  4545 (0-0)
22:59:37 03-FEB          0 Media Recovery Log
  /u02/mkslsb/lg01/mkslsbarch/log_4544_1_798129445.arc
22:59:38 03-FEB          0 Media Recovery Log
  /u02/mkslsb/lg01/mkslsbarch/log_4545_1_798129445.arc
22:59:39 03-FEB          0 Media Recovery Waiting for thread 1
  sequence 4546 (in transit)
```

NOTE: server was patched and rebooted, continued where it left off

```
01:35:19 04-FEB          0 ARC3: Beginning to archive thread 1
  sequence 4546 (10908582304950-10908640604514)
01:35:20 04-FEB          0 Media Recovery Waiting for thread 1
  sequence 4547 (in transit)
```

V$MANAGED_STANDBY

On Primary database

```
PROCES       SEQUENCE# STATUS           DELAY_MINS
------ ------------- ---------------- ----------
ARCH              4573 CLOSING                   0
ARCH              4574 CLOSING                   0
LNS               4575 WRITING                   0
```

On Standby database

```
PROCES       SEQUENCE# STATUS           DELAY_MINS
------ ------------- ---------------- ----------
ARCH                 0 CONNECTED                 0
RFS               3777 RECEIVING                 0
ARCH              4573 CLOSING                   0
ARCH              4574 CLOSING                   0
MRP0              4575 APPLYING_LOG              0
```

APPLYING_LOG - Process is actively applying the archived redo log to the standby.

WAIT_FOR_GAP - Process is waiting for the archive gap to be resolved