Data Guard: Setup and Administration

By

Ashok Kapur

Hawkeye Technology, Inc.

Agenda

- Introduction
- New Features
- Logical and Physical Standby Databases
- Setup Physical and Logical Standby Databases
- Administration including Switchover and Failover
- Best Practices
- Questions

Introduction

- Enhanced Standby Database Capabilities
- Increased Database Availability
- Increased Data Protection
- Part of Disaster Recovery Plan
- Offers offloading reporting functionality

New Features

New in Oracle 8i

- Open Standby in Read-Only Mode
- Using RMAN to backup Primary db using Standby db

New in Oracle 9.0.1

- Renaming of Standby Database to Data Guard
- Data Guard Broker and Data Guard Manager
- No Data Loss and No Data Divergence Features
- Built-in Database Switchover/Switchback
- Auto detection of Archive gaps

New in Oracle 9.0.1

- Auto add datafiles and Logfiles
- Parallel Recovery
- More Archive Destinations (upto 10)
- Standby Redo Logs
- ARCn available on Standby Database
- Delayed log application
- Various new init params, views and enhanced syntax for SQL cmds

New in Oracle 9.2

- Logical Standby Database
- Database protection modes
- Cascading Standby Databases
- More views

Physical and Logical Standby Databases

Physical Standby Database

- Same as the Standby Database in Oracle 8i:
  - In Managed Recovery Mode
  - In Read-Only Mode
- Is Physically Identical to the Primary db
- Supports DDL and DML commands for all data types.
- Great for Disaster Recovery and Data Protection

Logical Standby Database

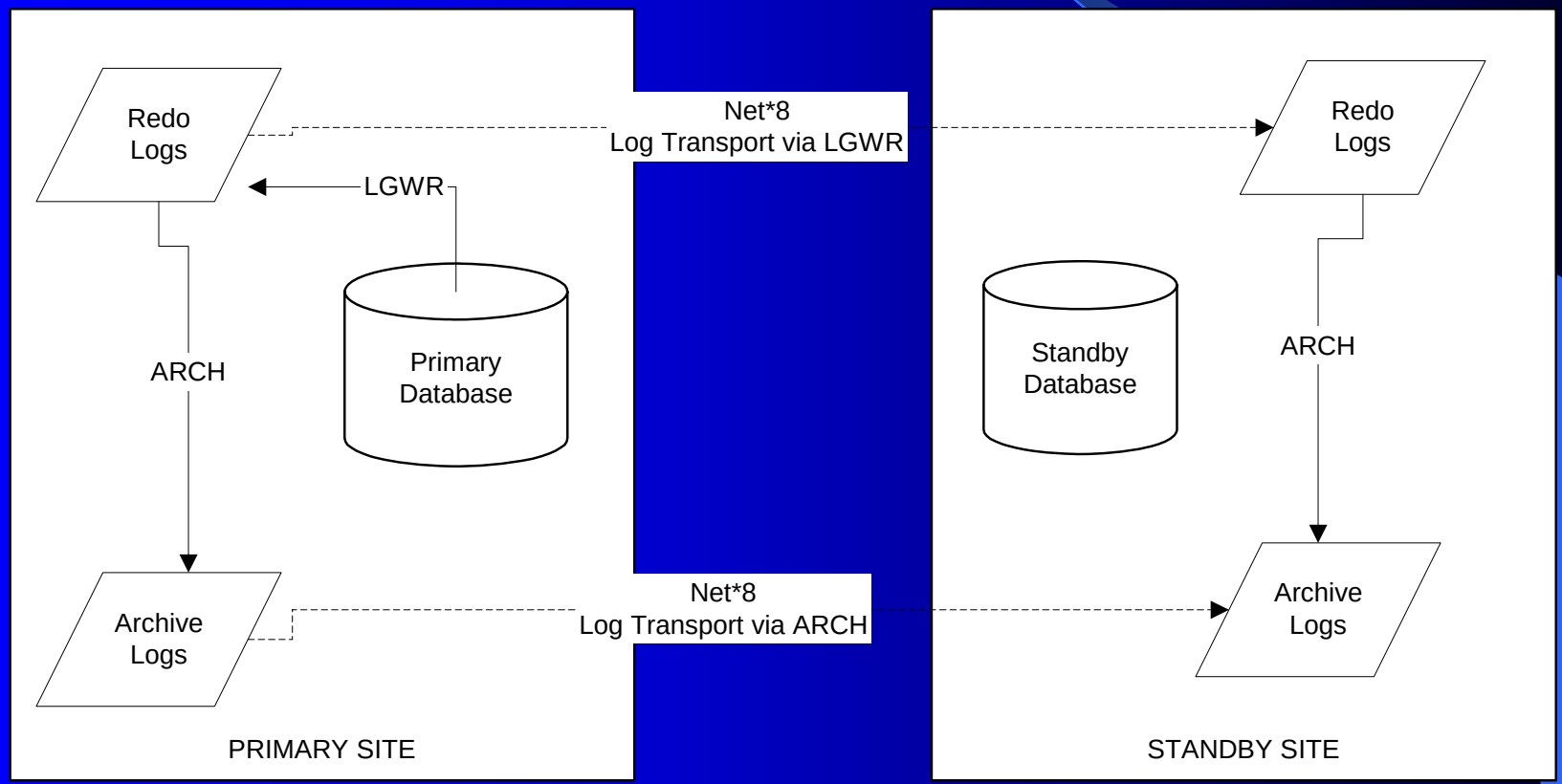
- Generally is not physically identical to Primary Database
- Can add additional objects: Indices, Materialized views,…
- Can exclude objects or schemas
- Is OPEN during recovery
- Uses LogMiner technology for applying logs

Logical Standby Database

- Unsupported Data Types:
  - NCLOB, LONG, LONG RAW, BFILE, UROWID, User Defined Types incl: Object Types, REFs, Varrays and Nested Tables
- Great as a Reporting Database
  - Offload Reporting Functionality

Setup Standby Database

Architecture



Standby Setup: Overview

1. Backup Primary DB
2. Ship backup to Standby Site
3. Add Log Transport on Primary
4. Add Net*8 settings on Primary
5. Setup Standby database
6. Setup Net*8 on Standby Site
7. Start Recovery

Create a Physical Standby Database

Pre-Requisites

- Primary Database in ARCHIVE LOG mode
- To prevent against NOLOGGING activities, setup Primary database to FORCE LOGGING
  - Alter database force logging;

Checklist

Steps	Description	Site
1	Make Sure Primary Database is in ARCHIVE Log mode and is set to Auto-Archive	Primary
2	Create a Backup of Primary Database datafiles	Primary
3	Create a Standby Control file	Primary
4	Create a Copy of init.ora file for Standby database.	Primary
5	Ship backup data files, standby control file and standby init.ora file to standby site	Primary
6	Setup Net*8 on Primary site to access Standby site	Primary
7	Setup Net*8 on Standby site to access Primary site	Standby
8	Make Primary and Standby init.ora changes	Standby
9	Startup Standby Database instance in recovery mode	Standby
10	Create Standby redo log files, if applicable	Standby
11	Setup instance parameters to ship redo log/archive log to standby site	Primary
12	Verify shipping and application of archive log	Primary/ Standby

Database Protection Modes

- Maximize Protection
  - No Data Loss and No data divergence
  - Arch_dest: mandatory, lgwr, sync, affirm
  - Primary db shutdown when unable to access stdby
- Maximize Availability
  - Arch_dest: mandatory, lgwr, sync, affirm
  - Protection auto lowered when stdby is unavailable
- Maximize Performance
  - Arch_dest: lgwr/arch, sync/async, mandatory/optional
  - Minimal performance impact

Detailed Steps

1. Verify primary db is ARCHIVE LOG mode
   - Select * from v$database; or
   - Archive log list;
2. Take a consistent backup of Primary db
3. Create standby control file
   - Alter database create standby controlfile;
4. Create a copy of primary db init.ora file
5. Ship backup data files, standby controlfile and init.ora file to Standby Site

Net*8 changes on Primary Site

- Listener.ora

```
STDBY = (DESCRIPTION =
   (ADDRESS = (PROTOCOL =    TCP)(HOST=primesite)(PORT =
     1522)))
SID_LIST_STDBY =
  (SID_LIST = (SID_DESC =
     (ORACLE_HOME = /u01/oracle/product/9.2.0) (SID_NAME =
      PRD01)))
```

- Tnsnames.ora

```
Stdbysite = (DESCRIPTION = (ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = stdbysite) (PORT = 1522))
(CONNECT_DATA = (SERVICE_NAME = stb01)))
```

Net*8 changes on Standby Site

- Listener.ora

```
STDBY = (DESCRIPTION =
   (ADDRESS = (PROTOCOL =    TCP)(HOST=stdbysite)(PORT =
     1522)))
SID_LIST_STDBY =
  (SID_LIST = (SID_DESC =
     (ORACLE_HOME = /u01/oracle/product/9.2.0) (SID_NAME =
      STB01)))
```

- Tnsnames.ora

```
Primsite = (DESCRIPTION = (ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = primsite) (PORT = 1522))
(CONNECT_DATA = (SERVICE_NAME = prd01)))
```

Init.ora Changes to Primary Db

- Setup Log Archive Destination
  - LOG_ARCHIVE_DEST_2 = ‘SERVICE=stdbysite OPTIONAL REOPEN=20’
  - LOG_ARCHIVE_DEST_2=ENABLE
  - LOG_ARCHIVE_FORMAT=arch%t%s.log
  - LOG_ARCHIVE_START=TRUE
  - REMOTE_ARCHIVE_ENABLE=TRUE

LOG_ARCH_DEST parameters

- [NO]AFFIRM
- ARCH/LGWR
- ASYNC=blocks
- [NO]DELAY[=minutes]
- MANDATORY/OPTIONAL
- SYNC

Init.ora Changes to Standby db

- Setup Log Archive Destination
  - LOG_ARCHIVE_DEST_1='LOCATION=/u03/oradata/arch'
  - LOG_ARCHIVE_DEST_1=ENABLE
  - LOG_ARCHIVE_FORMAT=arch%t%s.log
  - LOG_ARCHIVE_START=TRUE
  - REMOTE_ARCHIVE_ENABLE=TRUE
  - FAL_SERVER=primsite
  - FAL_CLIENT=stdbysite
  - DB_FILE_NAME_CONVERT=("/prd01", "/std01")
  - LOG_FILE_NAME_CONVERT=("/prd01", "/std01")
  - STANDBY_ARCHIVE_DEST=/u03/oradata/arch
  - STANDBY_FILE_MANAGEMENT=AUTO

Startup Standby Database

- Startup Standby Database:
  - Startup nomout;
  - Alter database mount standby database;
- Create Standby redo logs, if necessary:
  - Alter database add standby logfile group 5 (‘/u03/oradata/stb01/stbredo_51.log’, ‘/u03/oradata/stb01/stbredo_52.log’) size 10m;
  - Alter database add standby logfile group 5 (‘/u03/oradata/stb01/stbredo_61.log’, ‘/u03/oradata/stb01/stbredo_62.log’) size 10m;
- Recover managed standby database disconnect;

Final Steps

- Activate archive destination on Primary Database
- Set database protection
  - Alter database set standby to maximize performance;
- Verify the process:
  - On Primary db:
    - Alter system switch logfile
  - On Standby db:
    - Review receipt of archive log file
    - Review alert log to verify application of log file

Create a Logical Standby Database

Pre-Requisites

- Primary database running in ARCHIVELOG mode
- Determine whether primary db contains unsupported datatypes:
  - Select * from dba_logstdby_unsupported;
- Ensure tables have primary keys
  - If not, create disabled RELY constraints on the table
    - Alter table tabname add primary key(col1,col2) rely disable;

Pre-Requisites

- Enable supplemental logging on primary db
  - Select supplemental_log_data_pk, supplemental_log_data_ui from v$database;
  - Alter database add supplemental log data (primary key, unique index) columns;
- Start Resource Manager if using Hot backup for standby database
  - Alter system set resource_manager_plan=system_plan;
  - Restart database

Pre-Requisites

- Move LogMiner objects to alternate tablespace
  - Create tablespace logmnrtbs datafile ‘/u05/oradata/prd01/logmnrtbs01.dbf’ size 50m;
  - Exec dbms_logmnr_d.set_tablespace(‘logmnrtbs’);

Checklist

Steps	Description	Site
1	Create a Backup of Primary Database datafiles	Primary
2	Build LogMiner Dictionary	Primary
3	Create a Backup Control file	Primary
4	Create a Copy of init.ora file for Standby database.	Primary
5	Ship backup data files, backup control file and standby init.ora file to standby site	Primary
6	Setup Net*8 on Primary site to access Standby site	Primary
7	Setup Net*8 on Standby site to access Primary site	Standby
8	Make Primary and Standby init.ora changes	Standby
9	Startup Standby Database instance	Standby
10	Create data links from primary to stdby and from stdby to primary	Standby
11	Setup instance parameters to ship redo log/archive log to standby site	Primary
12	Verify shipping and application of archive log	Primary/ Standby

Detailed Steps

- Backup data files: cold backup or hot backup
- Note current SCN
  - Select checkpoint_change# from v$database;
- Build LogMiner Dictionary
  - Exec dbms_logstdby.build;
- Switch logfile and capture resultant archivelog
- Copy data files, controlfile and archivelog to stdby db
- Make init.ora changes on Standby db
  - Controlfile location
  - Standby_archive_dest
  - Log_archive_format

Startup Logical Standby Db

- Startup mount exclusive;
- Rename datafiles
- Clear logfile groups to create them
  - Alter database clear logfile group x;
- Recover database until SCN noted earlier
- Turn on Database Guard
  - Alter database guard [standby|all|none];

Startup Logical Standby Db

- Startup Standby database
  - Alter database open resetlogs;
  - Shutdown;
  - Startup mount exclusive;
- Use nid utility to change database name
  - Nid target=sys/syspwd dbname=log01 setname=yes
- Change DBNAME init.ora parameter to match new db name

Startup Logical Standby Db

- Startup db with reset logs
  - Startup mount exclusive;
  - Alter database open;
- Drop existing temp files and create new ones
  - Select * from v$tempfile;
  - Alter database tempfile '….' drop;
  - Alter tablespace tmptbs add tempfile '…'…;

Activate Standby Database

- Register the manually copied archive log
  - Alter database register logical logfile ‘/u02/oradata/arch/arch1251.log’;
- Apply the log giving the starting SCN # obtained in an earlier step
  - Alter database start logical standby apply initial <scn#>;

Final Steps

- Enable archiving to the standby site
- Verify the process:
  - On Primary db:
    - Alter system switch logfile
  - On Standby db:
    - Review receipt of archive log file
    - Review alert log to verify application of log file

Post-Processing

- On Primary db create dblink to standby db
- Bypass DataGuard
  - Exec dbms_logstdby.guard_bypass_on;
- On Standby db create dblink to primary db
- ReEnable DataGuard
  - Exec dbms_logstdby.guard_bypass_off;

Standby Administration

Adrninister Physical Standby Db

- Can backup standby database and use it to recover primary database
- Monitor DataGuard via:
  - v$archive_dest, v$archive_dest_status, v$archive_gap, v$archived_log,…
- Unlogged or Unrecoverable operations invalidate the Standby database
  - Direct Load
- Datafile rename
  - Manually rename file on Standby db

Adrninister Logical Standby

- Skip maintenance of unneeded tables or schemas
  - Skip all DML statements on actlog table
    - Alter database stop logical standby apply;
    - Exec dbms_logstdby.skip(‘DML’, ’APPL’, ’ACTLOG’, null);
    - Exec dbms_logstdby.skip(‘SCHEMA_DDL’, ’APPL’, ’ACTLOG’, null);
    - Alter database start logical standby apply;

Troubleshooting

- Review v$ views for errors
  - Log Transport from Primary
    - v$archive_dest_status – status of each destination
    - v$archived_log – review which archived logs have not been shipped to the standby destination
  - Log Application on Standby
    - v$managed_standby – status of log transport and log apply
    - v$archive_dest_status – archived_seq# vs applied_seq#
    - v$archived_log – list of arch logs received from primary
    - v$dataguard_status – Data Gurard related alert messages
    - dba_logstdby_progress – progress of SQL apply
    - v$archive_gap – any gaps in archive logs

Troubleshooting

- Review alert log
- Trace Data Guard processes
  - Set LOG_ARCHIVE_TRACE on either Primary or Standby or both
  - Alter system set log_archive_trace = <level>

Switchover and Failover

Switchover vs Failover

- Switchover
  - Gracefully switching roles of Primary DB to Standby DB and Standby DB to Primary DB
  - No Data Loss
  - Generally done for Os upgrades, etc.
- Failover
  - Switching Standby DB to Primary DB
  - Maybe data loss
  - Done when Primary DB crashes
  - Primary DB is lost and can not take over as Primary any longer

Switchover

Pre-Requisites

- Verify Primary site has updated Standby init.ora
- Verify Standby site has updated Primary init.ora
- Verify Standby site has caught up to the latest archive log file
- Standby database is running in ARCHIVELOG mode
- Verify all users have logged off the Primary db
- Verify possibility to switchover
  - Select switchover_status from v$database;

Physical Standby Switchover

Physical Standby Switchover

- Switchover Primary to Standby
  - Alter database commit to switchover to physical standby with session shutdown;
  - Shutdown;
  - Startup pfile=stdbyinit.ora nomount;
  - Alter database mount standby database;
- Verify database status on old Primary
  - Select switchover_status from v$database;
  - (Switchover Pending status)

Physical Standby Switchover

- Backup Standby Controlfile
  - Alter database backup controlfile to trace;
- Switchover Standby to Primary
  - Alter database commit to switchover to primary with session shutdown;
  - Shutdown
  - Startup pfile=prodinit.ora;
- Start Managed recovery on new Standby
  - Alter database recover managed standby database disconnect from session;

Physical Standby Switchover

- Start archiving on new primary db
  - Alter system archive log start;
- Verify Switchover success
  - On Primary:
    - Alter system switch logfile;
  - On Standby:
    - verify arrival of new archive log
    - Verify application of new archive log

Recover From Errors

- On old Primary db
  - Restore backup controlfile and startup database
- On Old Standby db
  - Restore backup controlfile and startup database
  - Start managed recovery
- Verify Data Guard Environment

Logical Standby Switchover

Logical Standby Switchover

- Switchover Primary to Standby
  - Alter database commit to switchover to logical standby;
  - (Don’t need to restart database.)
- Verify database status on new Standby (old Primary)
  - Select switchover_status from v$database;
  - (Switchover Pending status)

Logical Standby Switchover

- Switchover Standby to Primary
  - Alter database commit to switchover to primary;
  - Activate log_archive_dest_2 to send archive logs to new standby
- Start Managed recovery on new Standby
  - Alter database start logical standby apply new primary <dblink to new primary>;

Database Failover to Standby

Failover

- Failover is initiated when the Primary Database has crashed and it would take longer to recover Primary db than to perform Failover
- Failover will render Primary db incapable of becoming Standby db
- Standby db needs to be recreated from new Primary

Data Loss

- Data loss potential depends on Data Protection Mode of Data Guard
  - Maximum Protection: No Data Loss
  - Maximum Availability: Maybe no data loss
  - Maximum Performance: Data Loss

Failover Steps

- Physical Standby db
  - Alter database recover managed standby database finish;
  - Alter database commit to switchover to primary;
- Logical Standby db
  - Alter database activate logical standby database;
  - Alter database commit to switchover to primary;

Best Practices

Best Practices

- Create a Separate LISTENER for log transport
- Increase Net*8 Packet size for redo log transport
  - SDU parameter on both listener.ora and tnsnames.ora
  - Default Net*8 data pkt size 2K and default redo buffer size 1M
- Have at least one Physical Standby database for Failiver
- Pre-create redo log files on Phy Stdby db

Best Practices

- Regularly monitor log transport and application
- Regularly practice Switchover and Failover
- Pre-write Failover instructions for a non-DBA
- Know how to manually Failover/Switchover
- Have Copy of Standby init.ora on Primary and copy of Primary init.ora on Standby

Advanced Features/Further Study

- Multiple Standby Databases
- Standby Database in a RAC environment
- Cascading Standby Databases
- Archive only Standby Databases

Summary

Summary

- New Features
- Create Physical Standby Database
- Create Logical Standby Database
- Administer Standby Databases
- Switchover from Physical to Primary
- Switchover from Logical to Primary
- Best Practices

Questions?

References

- Oracle 9i SQL Reference
- Oracle 9i Utilities Guide
- Oracle 9i Net Admin and Reference Manuals
- Oracle 9i Data Guard Concepts and Administration
- Metalink Notes:
  - 150584.1,
  - 180031.1,
  - 219344.1,
  - 175122.1,
  - 185150.1

Contact

Ashok Kapur
Hawkeye Technology, Inc.
http://www.hawkeyetechnology.com
afkapur@hawkeyetechnology.com