Oracle 11*g* Data Guard Enhancements

By Inderpal Johal

DATA SOFTECH INC.
Complete Database Solution Provider

ORACLE CERTIFIED PARTNER

DATABASE
E-BUSINESS SUITE
Hyperion
SIEBEL

Oracle Certified Partner

AGENDA

- What is DATA GUARD ?
- Types of Data Guard Databases
- 11g Release 1 Enhancements
- 11g Release 2 Enhancements
- Active Data guard
- Snapshot Standby Database
- Role Based Services
- Rolling Upgrade and Physical Standby
- Q&A

11g DATAGUARD

- PHYSICAL STANDBY
  - ACTIVE DATA GUARD
  - SNAPSHOT STANDBY
- LOGICAL STANDBY
- ROLLING UPGRADE

11g DATAGUARD Introduction

- Performance with Distance over 100 kilometers
- Protection against Corruption using separate Databases
- Protection against User Errors using Option Delay

DATA GUARD - OVERVIEW

DATA GUARD => STANDBY DATABASE
PLUS MORE

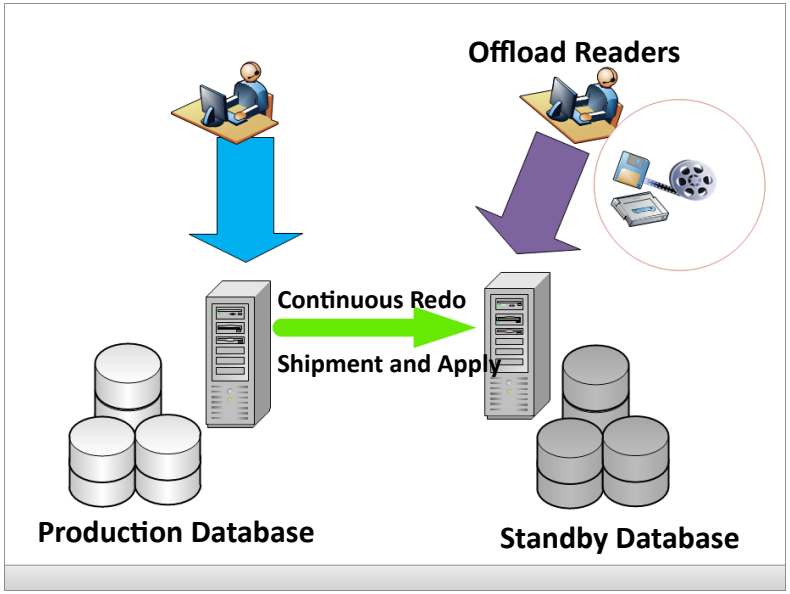
HIGH AVAILBILITY
&
DISASTER RECOVERY SOLUTION

- REDO TRANSPORT SERVICES
  - 
  - SYNC / ASYNC
  - AFFIRM / NOAFFIRM
- REDO APPLY SERVICES
  - 
  - REDO APPLY - PHYSICAL
  - SQL APPLY - LOGICAL
- ROLE MANAGEMENT SERVICES
  - 
  - PRIMARY ROLE
  - STANDBY ROLE
- ROLE TRANSITIONS
  - 
  - SWITCHOVER
  - FAILOVER
- DATA PROTECTION MODES
  - 
  - MAXIMUM PROTECTION
  - MAXIMUM AVAILABILITY
  - MAXIMUM PERFORMANCE

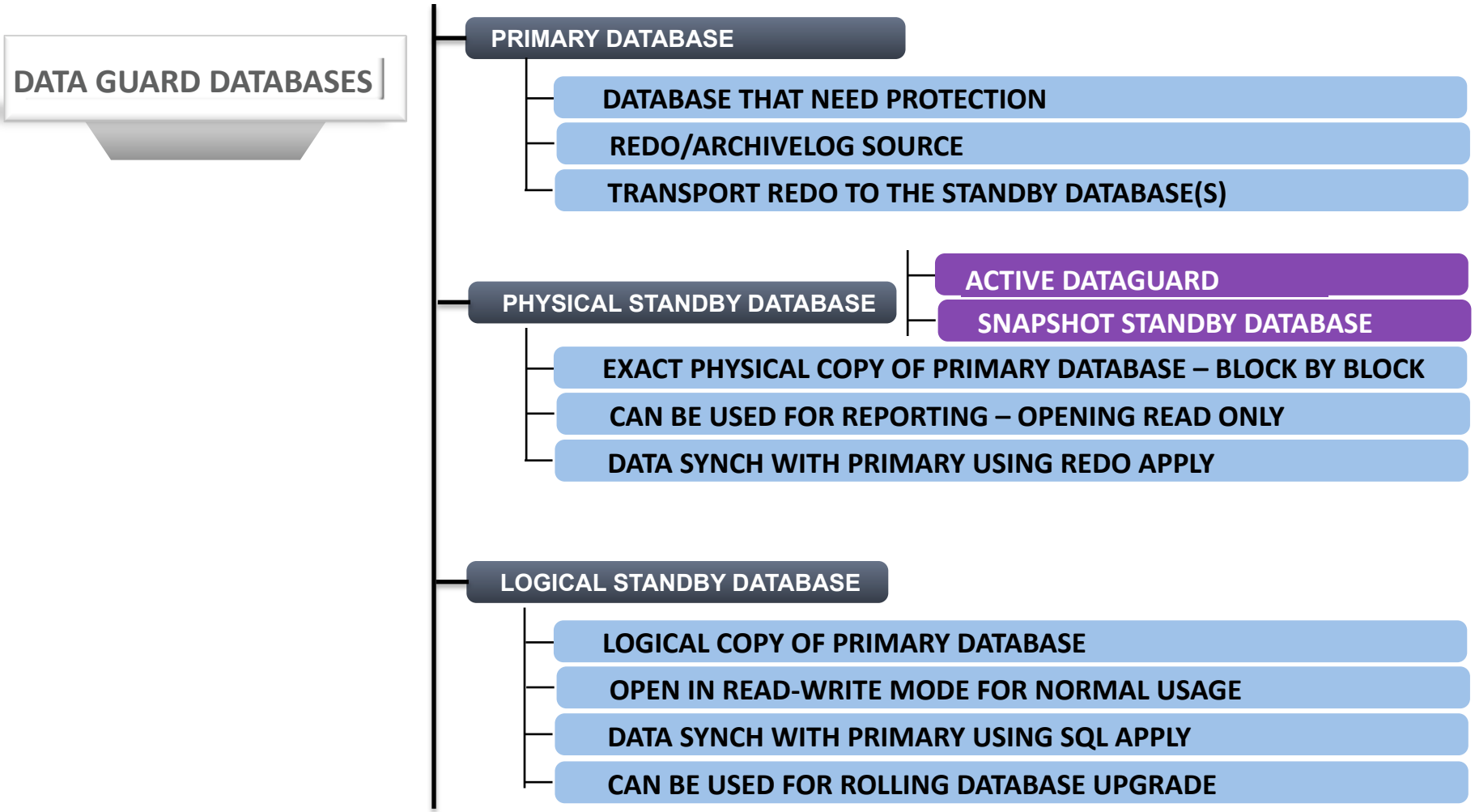
DATA GUARD PROVIDES

DATA GUARD

- HIGH AVAILABILITY
- DATA PROTECTION
- DISASTER RECOVERY FOR ENTERPRISE DATA
- CENTRALIZED STANDBY DATABASE MANAGEMENT



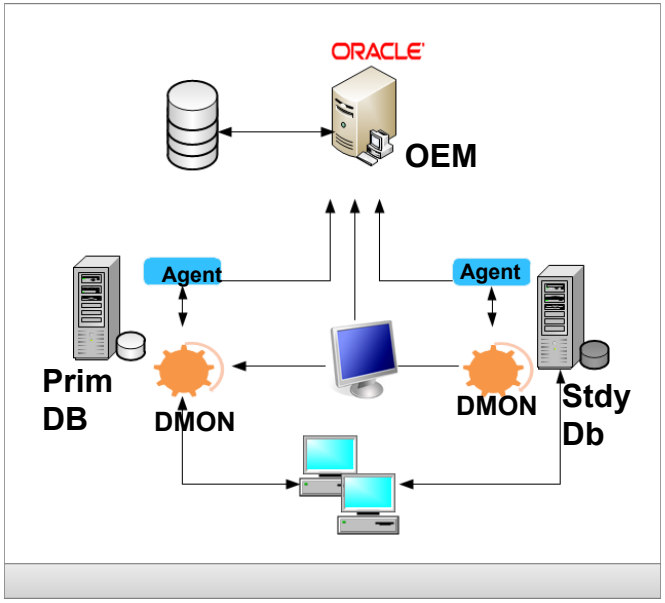
TYPES OF DATA GUARD DATABASES



11g DATAGUARD

- PHYSICAL STANDBY
  - ACTIVE DATA GUARD
  - SNAPSHOT STANDBY
- LOGICAL STANDBY
- ROLLING UPGRADE

11g DATAGUARD Rel 1
ENHANCEMENTS



11g RELEASE 1 - NEW FEATURES

11g Rel 1 FEATURES

- ACTIVE DATA GUARD
  - PHYSICAL STANDBY DATABASE OPENED READ-ONLY WITH APPLY
  - SIMALTANEOUS READ AND RECOVERY – REAL TIME QUERY
  - BLOCK CHANGE TRACKING FILE - FAST INCREMENTAL RMAN BACKUP
  - DATABASE OPTION FOR ORACLE ENTERPRISE EDITION
- SNAPSHOT STANDBY DATABASE
  - PHYSICAL STANDBY DATABASE OPENED READ-WRITE FOR TESTING
  - USED FOR UAT OR REAL APPLICATION TESTING
  - ARCHIVELOGS CONTINUE TO BE SHIPPED – PROVIDE DR PROTECTION
- RMAN ACTIVE DATABASE DUPLICATION
- REDO COMPRESSION – NEED ACO LICENSE
- FAST START FAILOVER IMPROVEMENTS
- HETEROGENOUS DATAGUARD SUPPORT
- LOST-WRITE DETECTION
- V$REDO_DEST_RESP_HISTOGRAM VIEW
- NO RESTART OF PRIMARY DATABASE
  - WHEN CHANGING FROM MAX-PERFORMANCE TO MAX-AVAILABILITY

11g RELEASE 2 - NEW FEATURES

11g Rel 2 FEATURES

- AUTOMATIC BLOCK RECOVERY
  - AUTOMATIC CORRUPT BLOCK RECOVERY USING PHYSICAL STANDBY
  - PHYSCIAL DATBASE MUST BE OPEN IN READ-ONLY WITH APPLY
  - NEED ACTIVE DATAGUARD
- ROLE BASED SERVICES
  - SERVICES ARE ASSOCIATED WITH ROLE OF THE DATABASE LIKE
    PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY | SNAPSHOT_STANDBY
  - SERVICES ARE MANAGED WITH GRID INFRASTRUCTURE
    ORACLE RESTART | ORACLE CLUSTERWARE
  - MUST BE CREATED ON BOTH PRIMARY AND STANDBY CLUSTERS
- NO RESTART OF PRIMARY DATABASE
  - WHEN CHANGING FROM MAX-AVAILABILITY TO MAX-PROTECTION
- SUPPORT UPTO 30 STANDBY DATABASES
  - LOG_ARCHIVE_DEST_1 …….. LOG_ARCHIVE_DEST_31 – PREVIOUSLY 9
  - COMPATIBLE =11.2.0 OR HIGHER FOR LOG_ARCHIVE_DEST_11--31
  - DEST_11—31 – Not to ALTERNATE, SYNC,MANDATORY, LOCATION

11g RELEASE 2 - NEW FEATURES

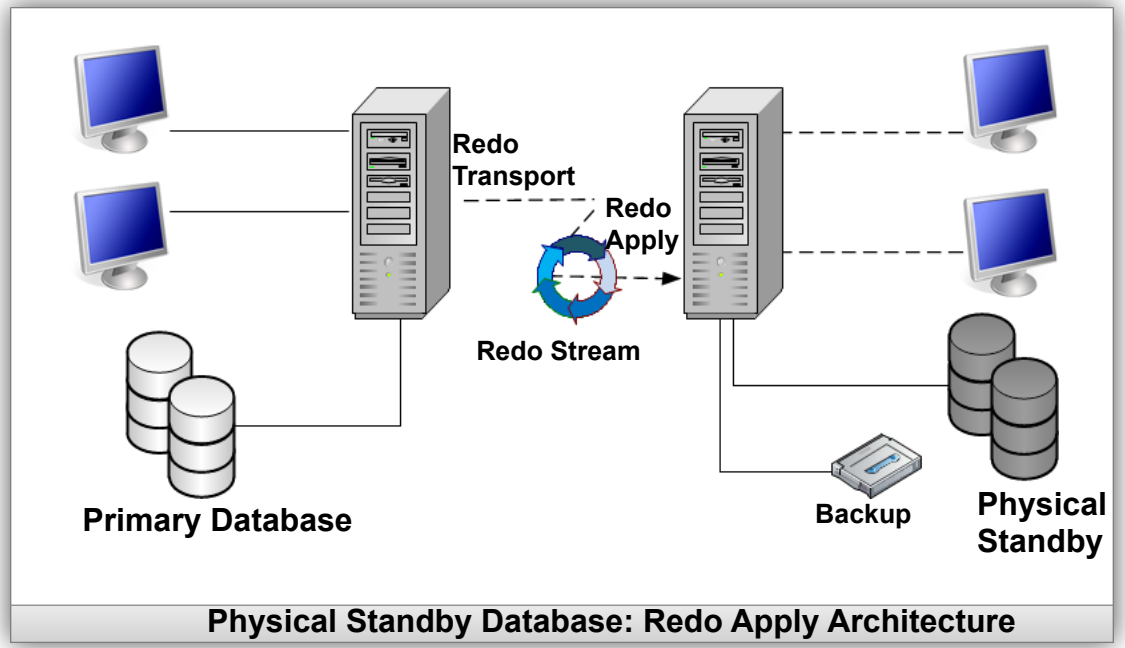
11g Rel 2 FEATURES

- ALTER SYSTEM FLUSH REDO – New COMMAND
  - FLUSH REDO FROM PRIMARY TO STANDBY DATABASE
  - USED IN FAILOVER OPERATION ON MOUNTED PRIMARY DATABASE
- ALTER SESSION SYNC WITH PRIMARY - New
  - WAIT FOR STANDBY TO BE SYNCHRONIZED WITH PRIMARY
  - FAILS IF REDO APPLY OR REDO TRANSPORT IS OFF
- STANDBY_MAX_DATA_DELAY – SESSION PARAMETER
  - USED FOR SESSION-SPECIFIC APPLY LAG TOLERANCE IN SECONDS
  - USED IN ACTIVE DATA GUARD ENVIRONMENT
- V$DATAGUARD_STATS
  - REPORT TRANSPORT AND APPLY LAGS
  - USED TO CHECK THAT APPLY LAG ON STANDBY IS IN LIMIT
- V$STANDBY_EVENT_HISTOGRAM VIEW
  - SHOW THE HISTORY OF APPLY LAG VALUES SINCE INSTANCE STARTUP

11g DATAGUARD

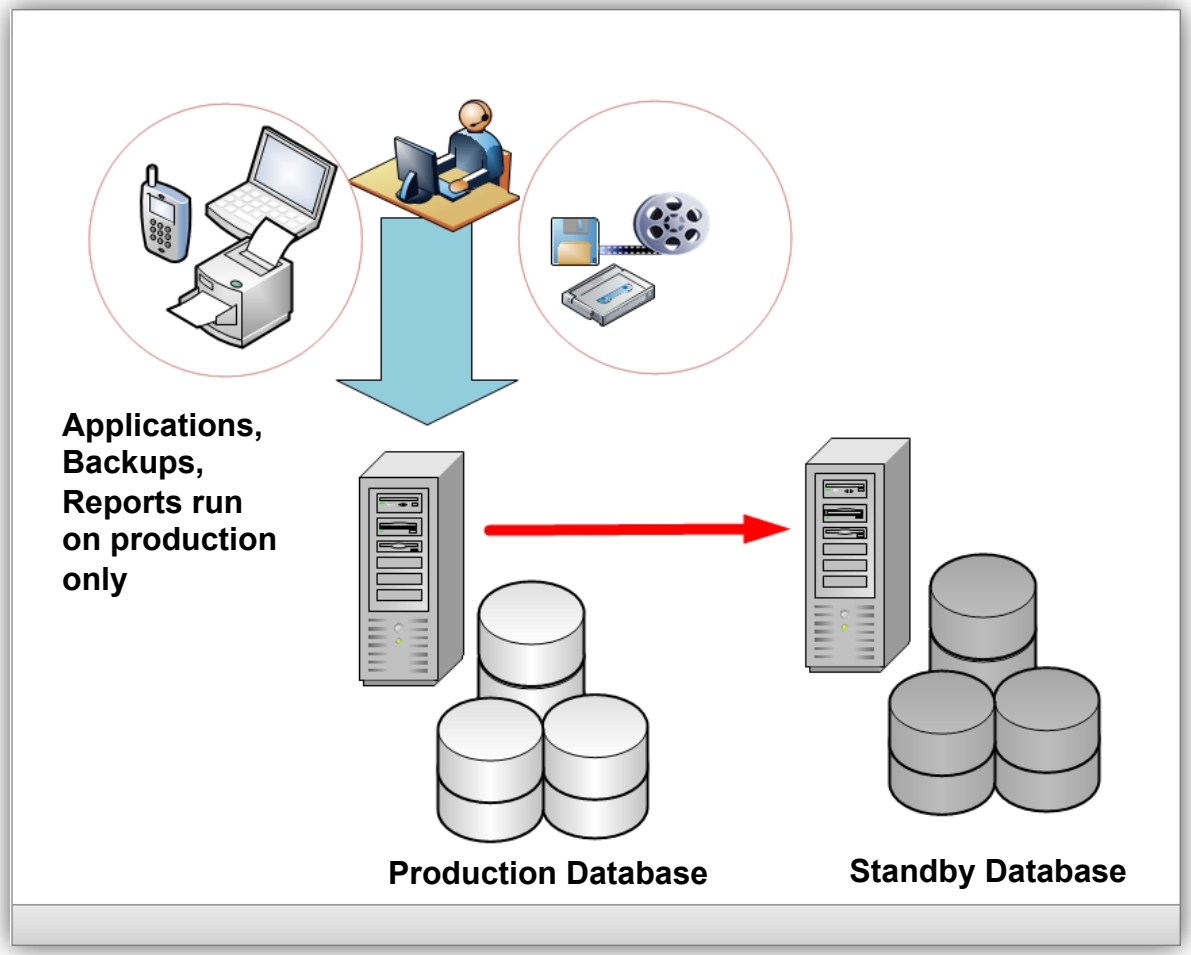
- PHYSICAL STANDBY
  - ACTIVE DATA GUARD
  - SNAPSHOT STANDBY
- LOGICAL STANDBY
- ROLLING UPGRADE

11g DATAGUARD
ACTIVE DATAGUARD

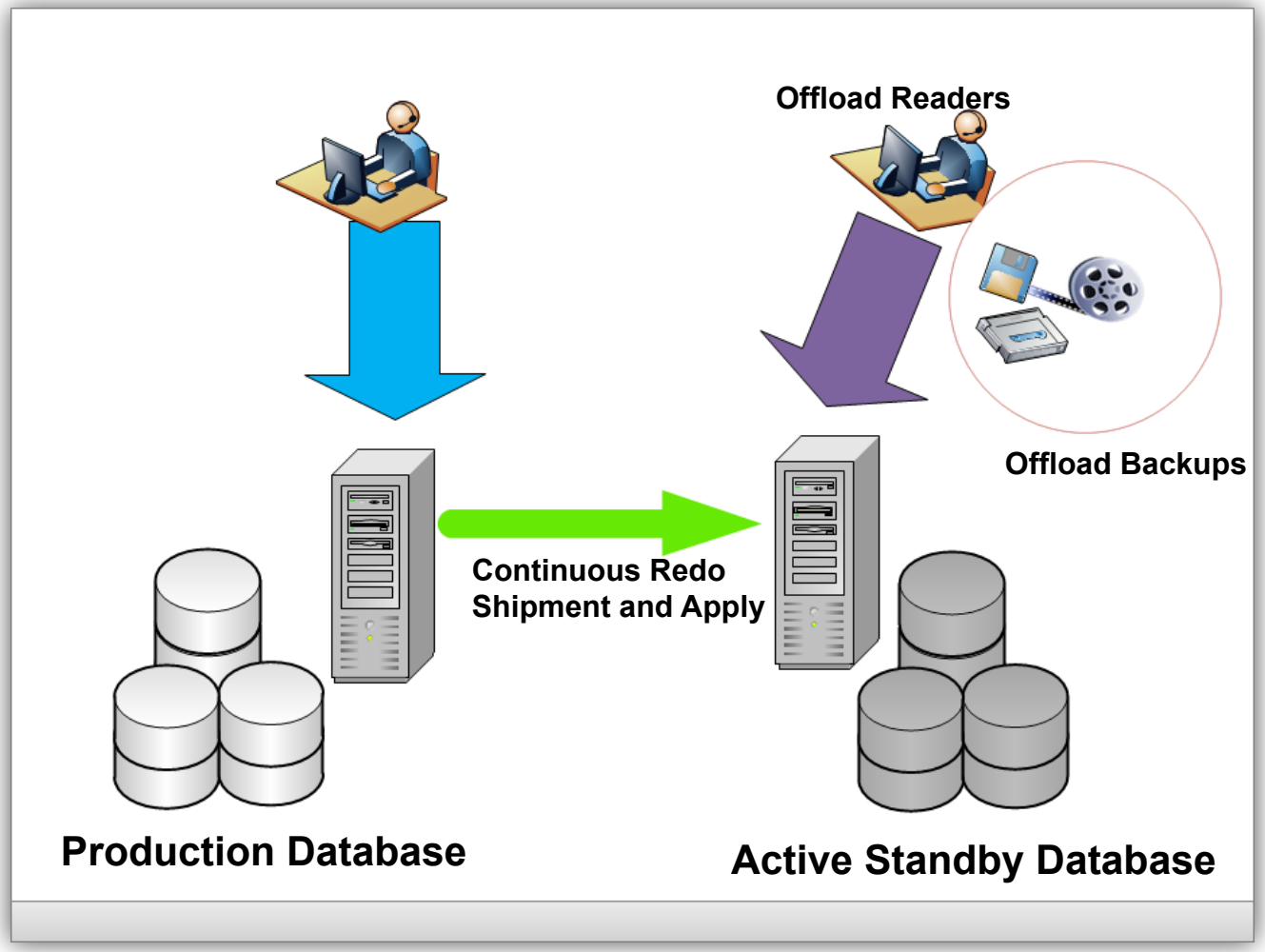


Physical Standby Database: Redo Apply Architecture

TRADITIONAL PHYSICAL STANDBY DATABASE USING DATA GUARD



USING ORACLE ACTIVE DATA GUARD



Offload Read-only queries to an up-to-date Physical Standby

Use Fast incremental backup on Active Physical Standby

Available with Enterprise Edition

Active Guard License is required

Guaranteed exact Replica of Production Data

ACTIVE DATA GUARD BENEFITS

Physical Standby in Recovery	→	Simultaneous Read and Recovery
Disaster Protection only	→	Performance Protection
Standby Server used Rarely	→	Real Time used as Production
All Read activity on Production	→	Move Read-Only Workload to Standby
Complex replication to create Replica	→	Simple setup and no Replica restriction
Need manual corrupted block recovery	→	Automatic Block Media Recovery
Backup load on Primary	→	Complete Incremental backup on standby

HOW TO TURN PHYSICAL STANDBY TO ACTIVE DATA GUARD ?

Stop Redo Apply – Cannot open database if redo apply is active

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

```
DGMGRL> EDIT DATABASE 'STDBY' SET STATE='APPLY-OFF';
```

Open the Physical Standby – Before starting redo apply

```
SQL> ALTER DATABASE OPEN READ ONLY;
SQL> SELECT name, database_role, open_mode FROM V$DATABASE;
NAME        DATABASE_ROLE          OPEN_MODE
STDBY       PHYSICAL STANDBY       READ ONLY
```

Restart Redo-Apply

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

```
DGMGRL> EDIT DATABASE 'STDBY' SET STATE='APPLY-ON';;
```

```
SQL> SELECT name, database_role, open_mode FROM V$DATABASE;
NAME        DATABASE_ROLE          OPEN_MODE
STDBY       PHYSICAL STANDBY       READ ONLY WITH
     APPLY
```

WHAT OPERATION ALLOWED IN ACTIVE DATA GUARD ?

- SELECT statements
- Complex queries
- Calling of stored procedures
- Use of database links to redirect DML to Primary
- Use of stored procedures to call remote stored procedures via database links
- Use of SET ROLE
- Use of ALTER SESSION and ALTER SYSTEM

- No Datatypes restrictions like Logical Standby
- Additional TEMPFILES can be created for Sorts
- Flashback Query : SELECT ... AS OF <SCN>

HOW WRITE OPERATION ALLOWED IN ACTIVE DATA GUARD ?

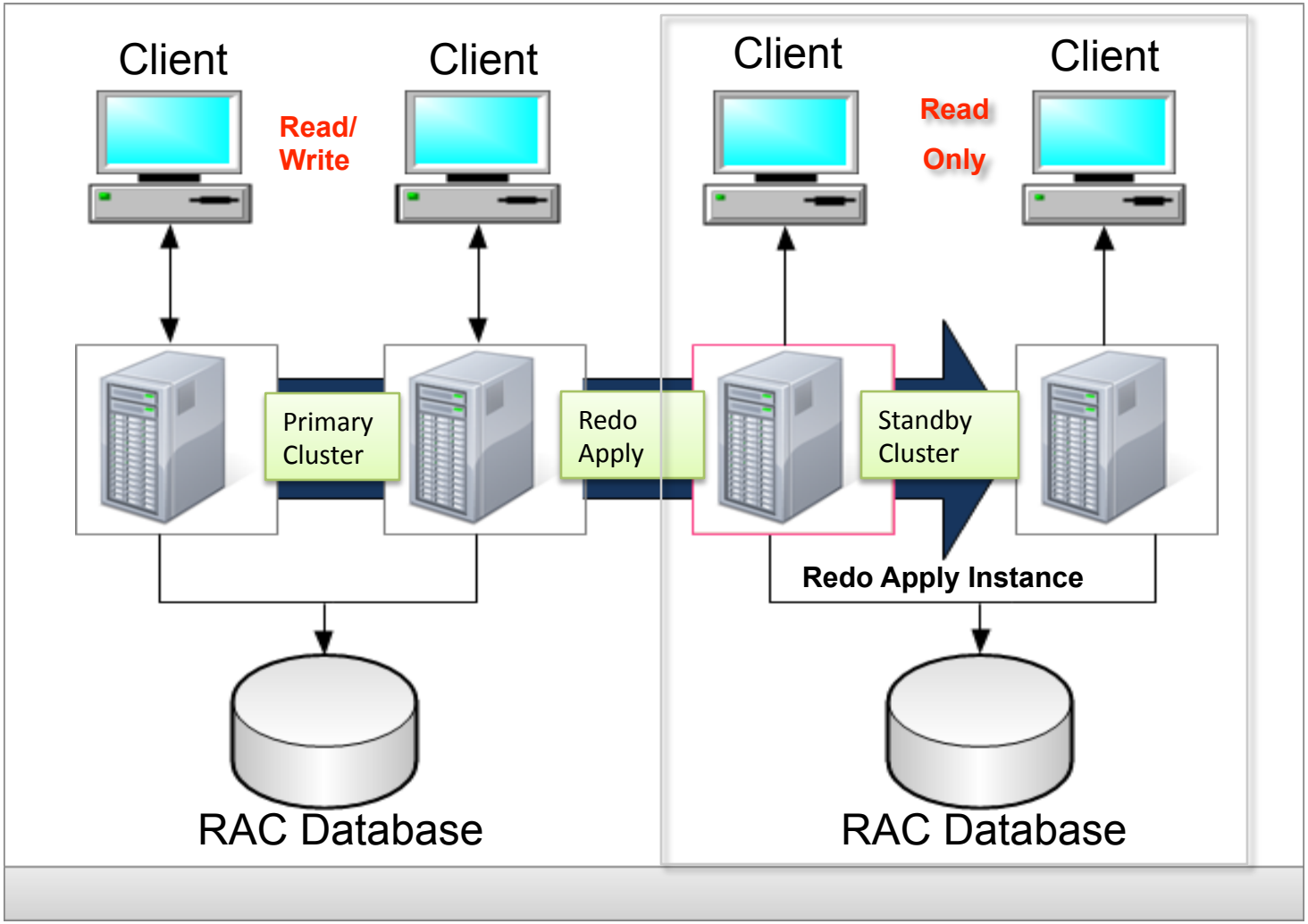
1. Create Database link in Primary Database for SCOTT user
   CREATE DATABASE LINK PRIM ..... USING 'prim'
2. Redo will Applied to Standby and so Database Link will be visible
3. Connect to Standby and Insert Data using Database Link
   INSERT INTO emp@PRIM VALUES .....
4. Data will be Added to Primary Database using Database Link
5. Redo from Primary will be applied to Standby Database
6. Data will be visible in Standby Database

TRANSPARENTLY REDIRECTING WRITES TO PRIMARY

- Create database link on Primary
- Creates User STD for Primary Application SCOTT and Creates Synonym for SCOTT
  - STD.Tablename for READ and STD.Tablename@dblink
- Application Connects to PRIMARY Database and Read and Write to user SCOTT Objects
- Reporting Application connects to Standby and do READs. All Writes will be directed to Primary Database using dblink

```
CREATE TRIGGER logon_switch_schema_trigger
  AFTER LOGON ON scott.schema
  BEGIN
    IF (SYS_CONTEXT('USERENV','DATABASE_ROLE')
        IN ('PHYSICAL STANDBY'))
    THEN
      execute immediate
        'alter session set current_schema = std';
    END IF;
  END;
```

ACTIVE DATA GUARD IN RAC ENVIRONMENT



UNDERSTANDING DATA LAG IN ACTIVE DATA GUARD DATABASE

- Active Data Guard can have data lag from the Primary Database because of :
  - High Network Latency
  - Limited Bandwidth
- Queries on the Standby Database need to provide Real-Time Current Results with some defined SLA
- Need Ways to manage Standby Lag and take appropriate action :
  - Configure Data Guard to trigger an error when Data Lag exceed SLA limit
  - Monitor the Redo Apply Lag and take action when the lag is unacceptable
- 11g R2 provides features to determine the Data lag and take appropriate action as shown below

11g FEATURES

- V$DATAGUARD_STATS
- V$STANDBY_EVENT_HISTOGRAM VIEW
- STANDBY_MAX_DATA_DELAY
- ALTER SESSION SYNC WITH PRIMARY
- V$REDO_DEST_RESP_HISTOGRAM VIEW

MONITORING APPLY LAGS ON STANDBY DATABASE

- Monitor apply lag for Real-time queries on Physical Standby database with new view V$DATAGUARD_STATS

```
SELECT name,value, datum_time, time_computed
FROM   V$DATAGUARD_STATUS
WHERE   name LIKE 'apply lag';
```

NAME	VALUE	DATUM_TIME	TIME_COMPUTED
Apply lag	+00 00:00:00	03/10/2011 11:11:11	03/10/2011 11:11:11

DATUM_LINE => Timestamp when data was last received by the Standby Database
TIME_COMPUTED => Timestamp taken when the apply lag metric was calculated
VALUE => Shows the Apply Lag in seconds

- Another view V$STANDBY_EVENT_HISTOGRAM is provided to track the apply lag since the standby database was last started

```
SELECT * FROM V$STANDBY_EVENT_HISTOGRAM
WHERE name = 'apply lag' AND count > 0;
```

NAME	TIME UNIT	COUNT	LAST_TIME_UPDATED
Apply lag	0 seconds	500	3/10/2011 11:11:11

ESTABLISHING SERVICE LEVEL FOR QUERIES ON STANDBY

- NEW Session level parameter STANDBY_MAX_DATA_DELAY and Default to NONE
- Specifies time limit in seconds allowed between when changes are committed on Primary and then visible on Standby database
- Ignored for the SYS user
- Warning is displayed to the client as well as in the Standby Alert Log
- A value of
  NONE => Queries issued against Physical Standby will be executed regardless of Apply lag
  Non-Zero => Queries issued against Physical Standby will be executed only if apply lag is less than or equal to STANDBY_MAX_DATA_DELAY
  => ORA-3172 error is returned to the client
  Zero => Queries issued against Physical Standby is guaranteed to return the exact same data as it is available on the Primary database
  => Make sure PROTECTION MODE must be :
  - Maximum Availability
  - Maximum Protection
  => Redo APPLY must be Active
  => If the Standby database is lagging behind then ORA-3172 will be returned

ORA-3172 STANDBY_MAX_DATA_DELAY has been exceeded

ENABLE STANDBY_MAX_DATA_DELAY FOR APPLICATION QUERY

- New DATABASE_ROLE attribute of the USERENV context
- The database role is one of the following: PRIMARY, PHYSICAL STANDBY, LOGICAL STANDBY, SNAPSHOT STANDBY

SELECT sys_context('USERENV', 'DATABASE_ROLE')
FROM dual;

```
CREATE OR REPLACE TRIGGER sla_trigger
   AFTER LOGON
   ON APP.SCHEMA
BEGIN
   IF (SYS_CONTEXT('USERENV', 'DATABASE_ROLE')
       IN ('PHYSICAL STANDBY'))
   THEN execute immediate
     'alter session set standby_max_data_delay=5';
   ENDIF;
END;
```

CONFIGURING ZERO LAG BETWEEN PRIMARY AND STANDBY

- NEW Session statement
  ALTER SESSION SYNC WITH PRIMARY;
- It will block all query against the standby until all Redo data received from PRIMARY has been applied on STANDBY database
- Once above ALTER Command is finished, session starts processing queries without waiting for standby redo apply
- ORA-3173 will be returned to all client if
  - Redo-apply is not active
  - ALTER command is cancelled before it is finished

```
SQL> CREATE OR REPLACE TRIGGER hr_logon_sync_trigger
  2 AFTER LOGON ON hr.schema
  3 BEGIN
  4  IF (SYS_CONTEXT('USERENV','DATABASE_ROLE')
  5     IN ('PHYSICAL STANDBY'))
  6  THEN
  7   execute immediate 'alter session sync with primary';
  8  END IF;
  9  END;
 10  /

Trigger created.
```

V$REDO_DEST_RESP_HISTOGRAM VIEW

Used to monitor the Synchronous Redo Transport Response Time

- Contains response time data for each redo transport destinations sent via the synchronous redo transport mode.
- Each Destination has series of rows with one row for each Response Time
- Response are rounded to nearest whole second of 300Sec for response less than 300 sec. Similarly response Greater than 300 s are rounded to 600, 1200, 2400, 4800 or 9600s
- This view will be helpful for
  - Identifying Sync Redo Transport issue affecting transaction throughput on primary
  - Can be used to Tune NET_TIMEOUT attribute

Display the slowest response time for destination 2:

```
SQL> SELECT max(DURATION)
       FROM V$REDO_DEST_RESP_HISTOGRAM
       WHERE DEST_ID=2 AND FREQUENCY>1;
```

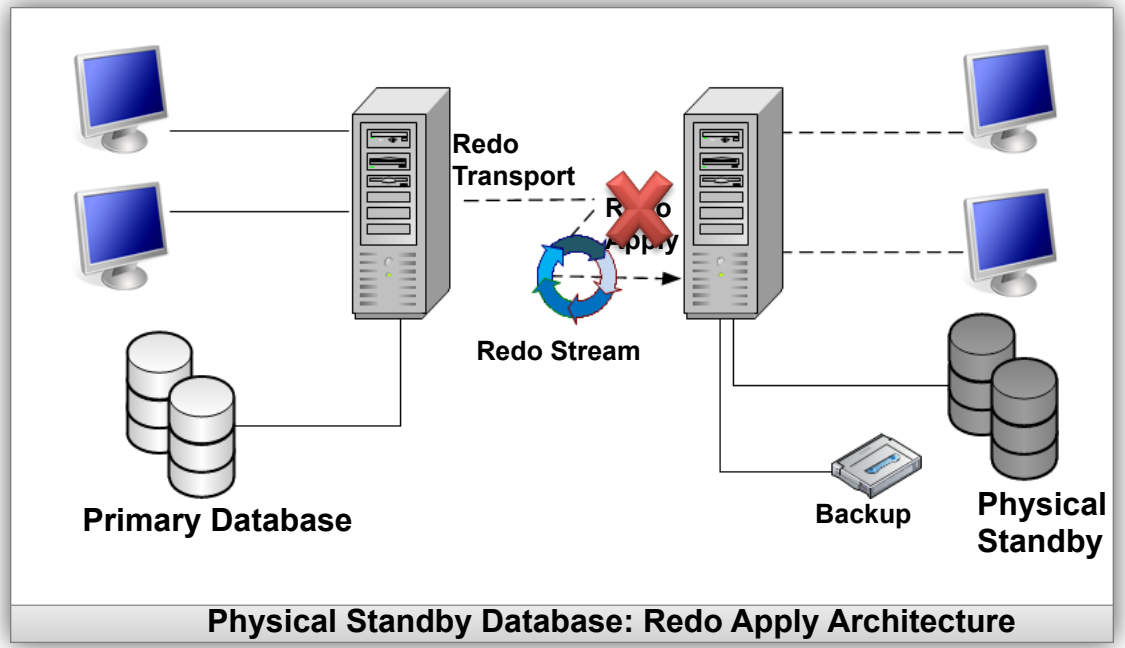
Display the fastest response time for destination 2:

```
SQL> SELECT min(DURATION)
       FROM V$REDO_DEST_RESP_HISTOGRAM
       WHERE DEST_ID=2 AND FREQUENCY>1;
```

11g DATAGUARD

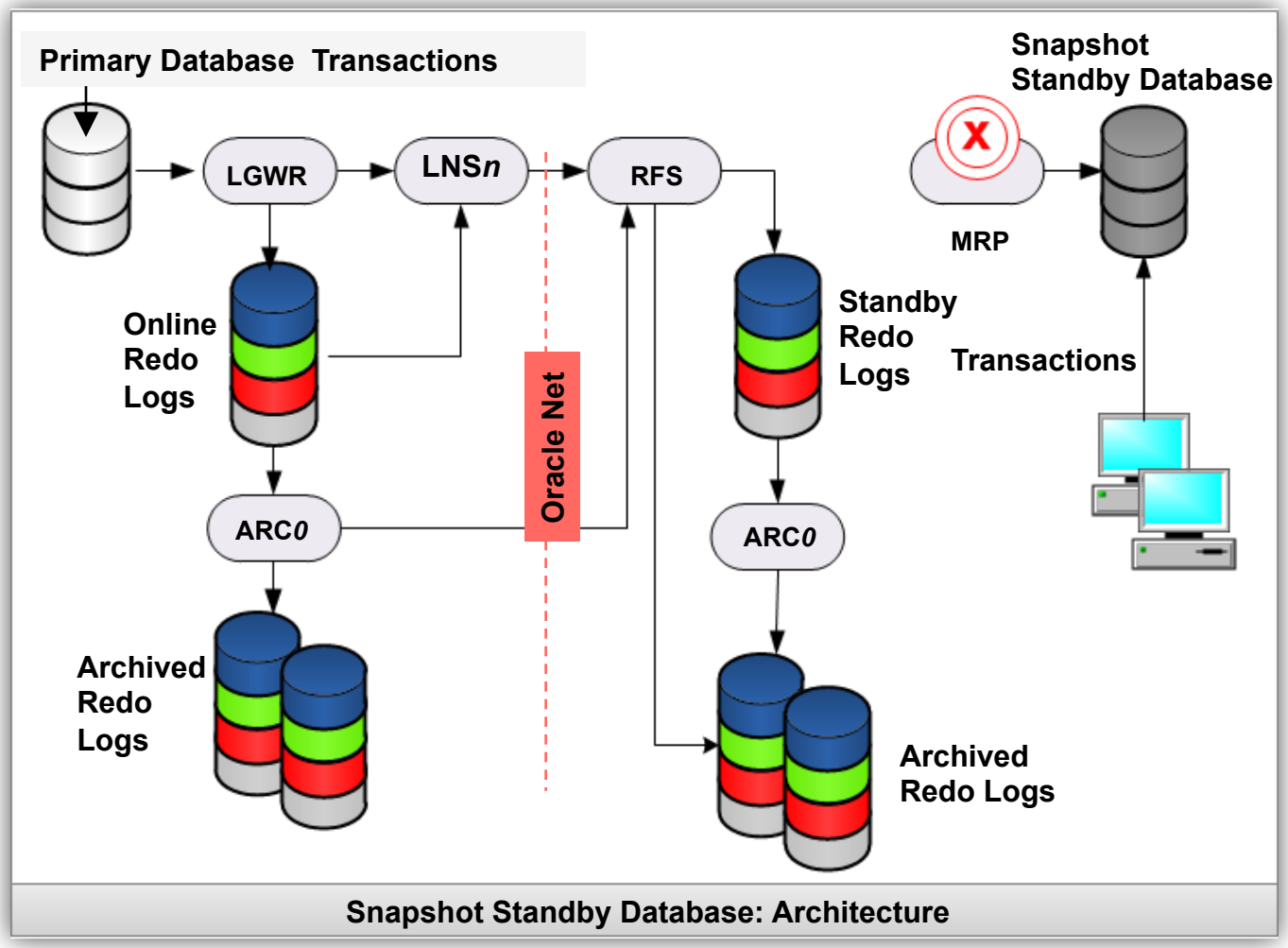
- PHYSICAL STANDBY
  - ACTIVE DATA GUARD
  - SNAPSHOT STANDBY
- LOGICAL STANDBY
- ROLLING UPGRADE

11g DATAGUARD
SNAPSHOT STANDBY



Physical Standby Database: Redo Apply Architecture

ORACLE SNAPSHOT STANDBY DATABASE



Snapshot Standby Database: Architecture

Fully updateable standby– Continuous redo transport while open Read-write

Provides Disaster Recovery and data protection

Truly leverages standby database hardware

No Extra License to use this option

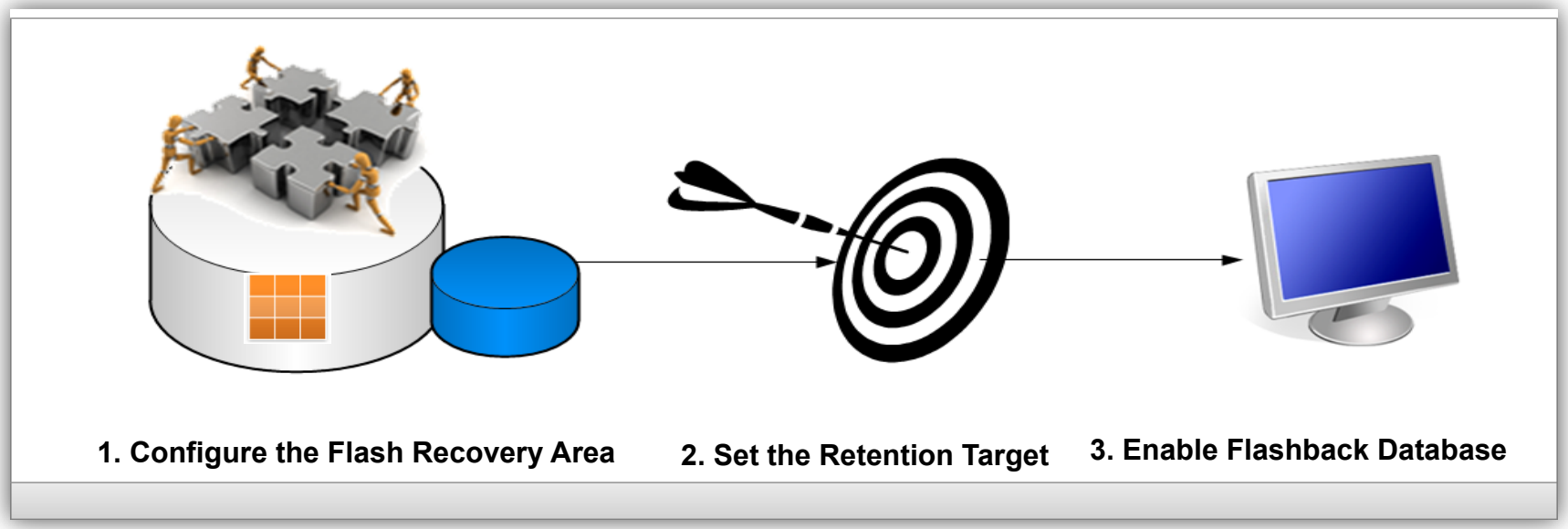
Use the following to convert Physical Standby to Snapshot Standby

1. DGMGRL
2. OEM
3. SQLPLUS

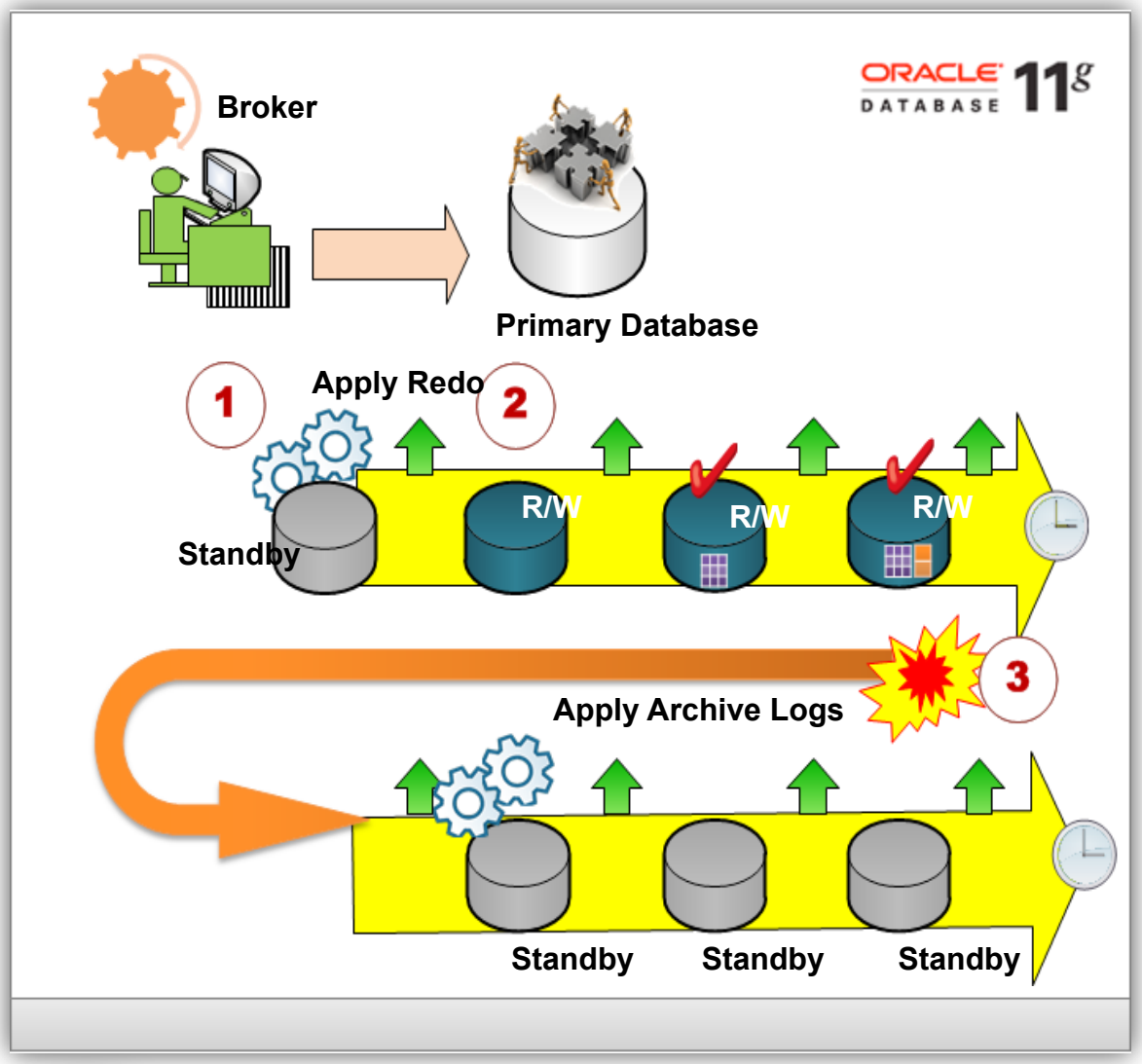
SOME IMPORTANT POINTS FOR SNAPSHOT STANDBY DATABASE

POINTS

- PHYSICAL STANDBY DATABASE SHOULD BE IN FLASHBACK MODE
- NO REAL TIME QUERY OR DATA
- NO FAST START FAILOVER IN THIS MODE
- NO SWITCHOVER/FAILOVER UNTIL CONVERTED TO PHYSICAL STANDBY



ORACLE 11g SNAPSHOT STANDBY DATABASE



Snapshot Standby

1. Convert to Snapshot Standby
2. Use Snapshot Standby
3. Convert to Physical Standby

ORACLE 11g SNAPSHOT STANDBY DATABASE

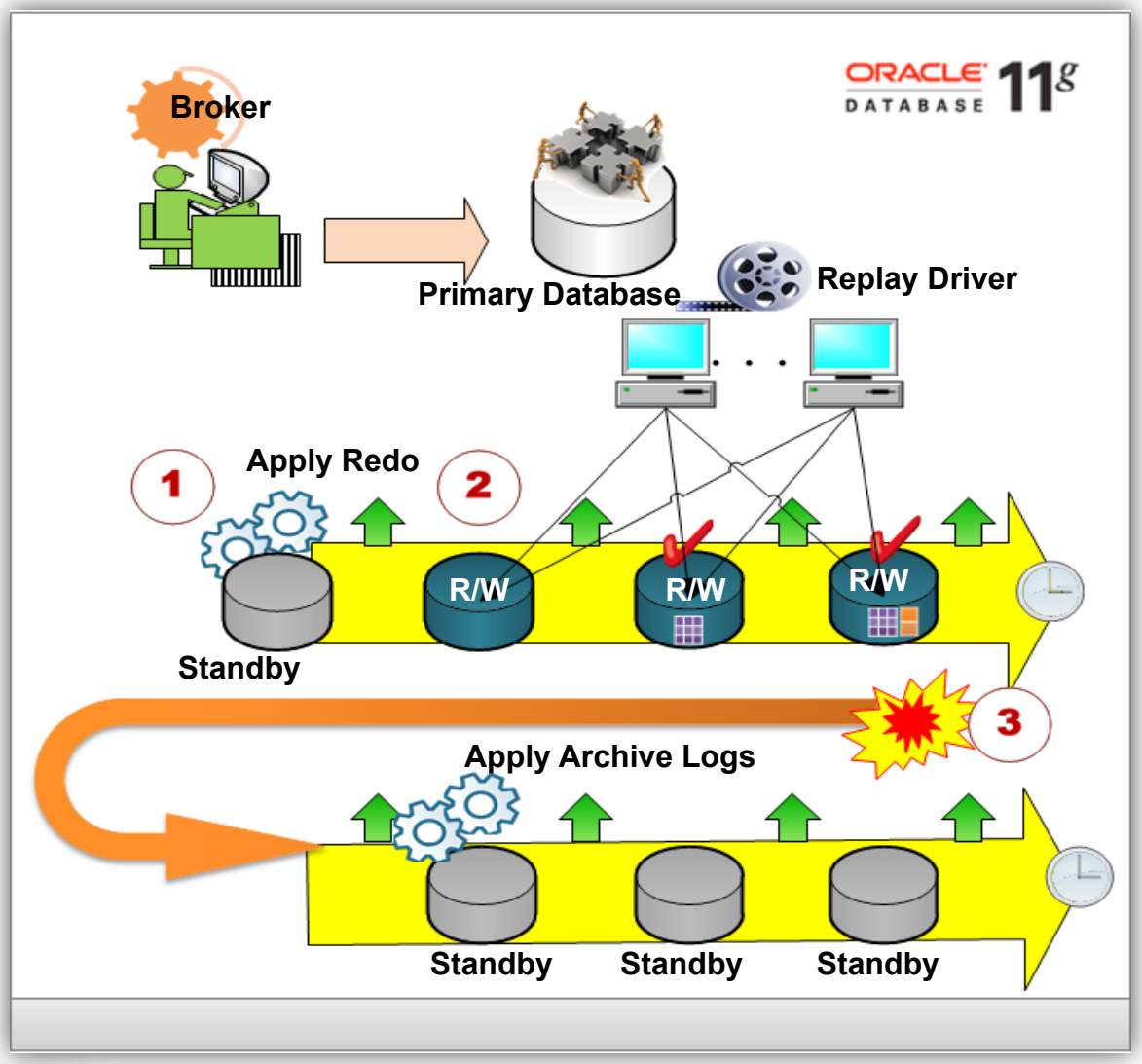
1. **Confirm that Flashback Database is enabled on the Physical Standby**
   SQL> SELECT FLASHBACK_ON FROM V$DATABASE;

2. **Convert the Physical Standby to Snapshot database.**
   DGMGRL> CONVERT DATABASE ‘stndby' TO SNAPSHOT STANDBY;

3. **Do all Testing as per the requirements**

4. **Convert Snapshot Standby to Physical Standby database**
   DGMGRL> CONVERT DATABASE ‘stndby' TO PHYSICAL STANDBY;

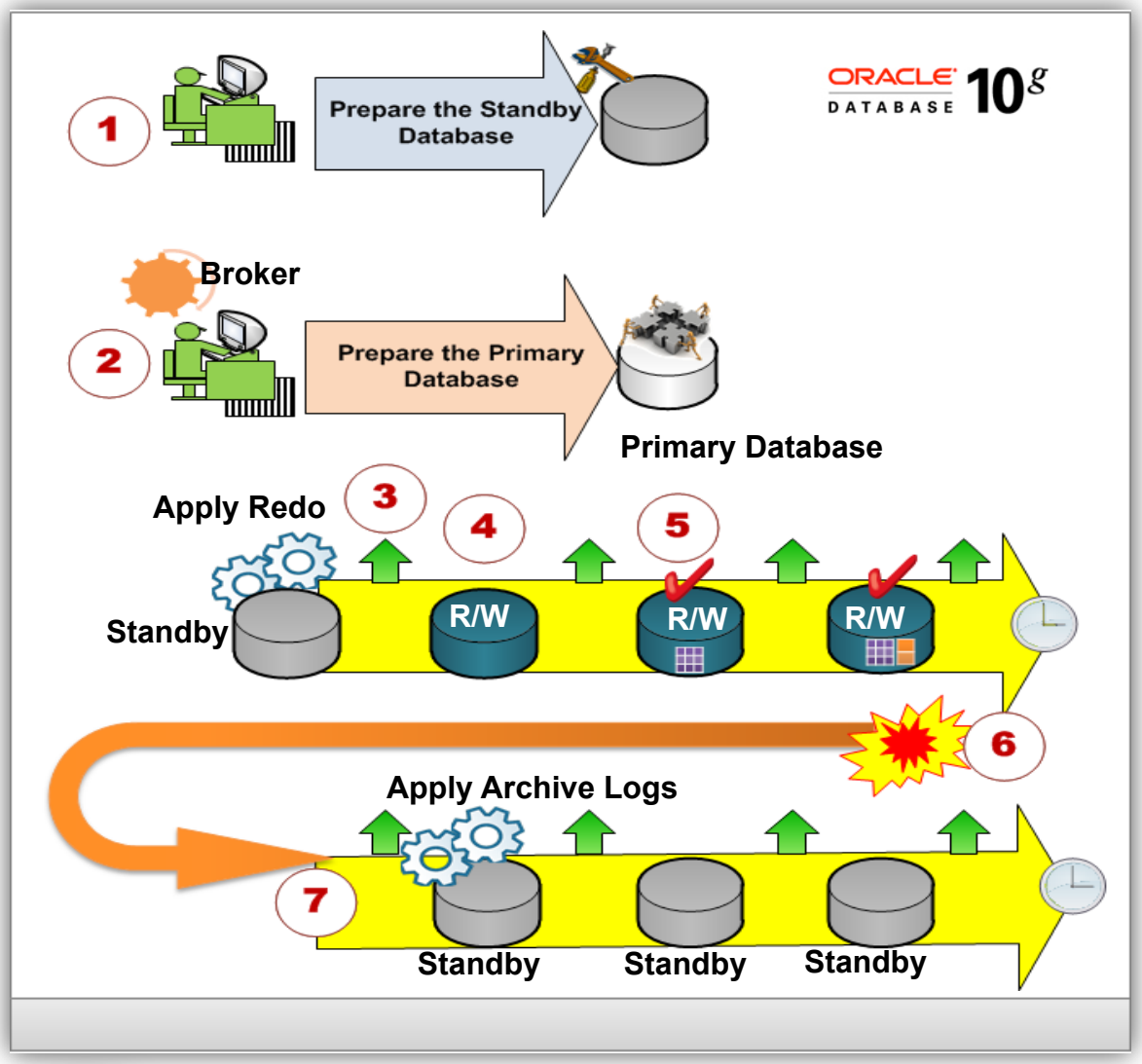
ORACLE 11g SNAPSHOT STANDBY DATABASE - RAT



Use with Real Application Testing

- Capture the Load on Primary using DB Replay/RAT
- Copy the Captured file to Standby Database Server
- 1. Convert to Snapshot Standby
- 2. Use Snapshot Standby
- 3. Convert to Physical Standby

SNAPSHOT STANDBY EQUIVALENT IN ORACLE 10g



Snapshot Standby

1. Prepare the Standby
2. Prepare the Primary
3. Create Guaranteed Restore Points
4. Convert Physical Standby to Read/ Write
5. Use Read Write Standby
6. Flashback to Restore Point
7. Convert Standby back to Physical Standby

SNAPSHOT STANDBY EQUIVALENT IN ORACLE 10g

1

Prepare the physical standby database to be activated.

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=20G;
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/arch/oradata'
```

2

Prepare the primary database to have the physical standby be diverged.
Archive the current log file.

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

3

Cancel Redo Apply and create a guaranteed restore point.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> CREATE RESTORE POINT restore_pt GUARANTEE FLASHBACK DATABASE;
```

4

Activate the physical standby database.

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
SQL> STARTUP MOUNT FORCE;
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
SQL> ALTER DATABASE OPEN
```

5

Revert the activated database back to a physical standby database.

```
SQL> STARTUP MOUNT FORCE;
SQL> FLASHBACK DATABASE TO RESTORE POINT restore_pt;
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
SQL> STARTUP MOUNT FORCE;
```

11g DATAGUARD

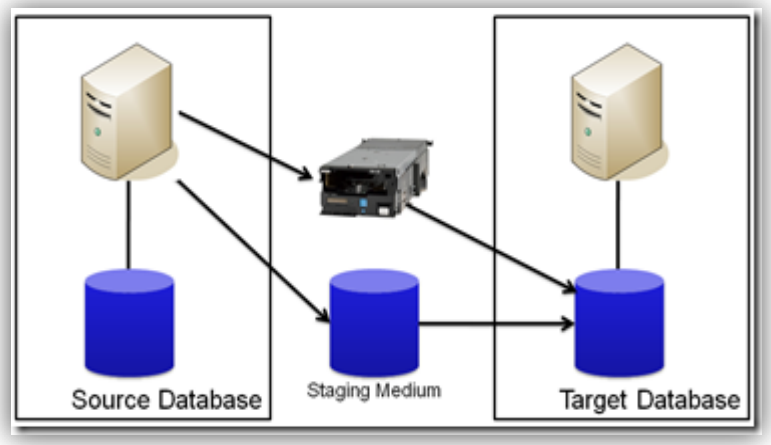
- PHYSICAL STANDBY
  - ACTIVE DATA GUARD
  - SNAPSHOT STANDBY
- LOGICAL STANDBY
- ROLLING UPGRADE

11g DATAGUARD MORE FEATURES

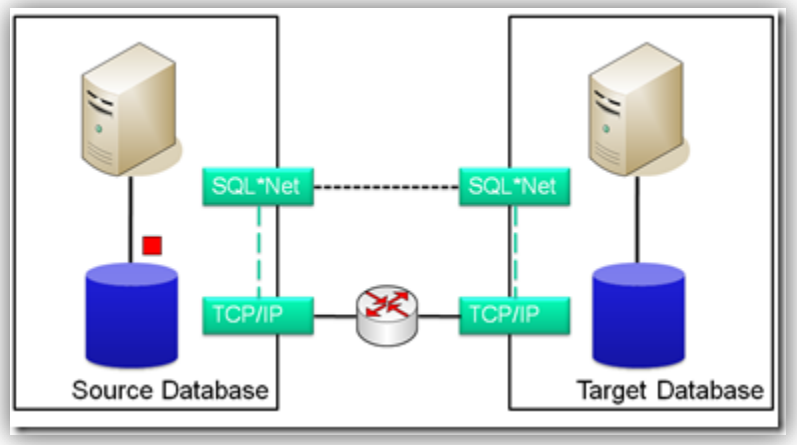
- RMAN STANDBY DUPLICATION
- REDO COMPRESSION
- HETEROGENOUS SUPPORT
- LOST WRITE DETECTION
- AUTOMATIC BLOCK RECOVERY
- ROLE BASED SERVICES
- FLUSH REDO

RMAN ACTIVE DATABASE DUPLICATION

- Goal is the creation of a duplicate database containing all or subset of source database
- Clone database from ACTIVE database instead of RMAN backup
- Automatically assign new DBID to register the new database in Recovery Catalog or so



Backup-based Duplication



Active Database Duplication

NEW RMAN SYNTAX FOR STANDBY DATABASE DUPLICATION

```
RMAN> connect target sys/oracle
RMAN> connect auxiliary sys/oracle@stndby
run {
        ALLOCATE CHANNEL prim1 TYPE disk;
   ALLOCATE CHANNEL prim1 TYPE disk;
     ALLOCATE AUXILIARY CHANNEL stdby TYPE DISK;

   DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE ;
spfile
   set db_unique_name='stndby'
   ....
   ....
    }
```

DATA GUARD REDO COMPRESSION

- Redo Compression
  - Compress Redo data as it is transmitted to standby
  - Reduce Bandwidth consumption but Increases CPU utilization

- Automatically compresses data transmitted only to resolve GAPS ➔ Oracle 11g R1

FASTER GAP RESOLUTION ➔ BETTER DATA PROTECTION

- Very helpful in bandwidth constrained environments

REDUCE TRANSMISSION TIME ➔ 15 – 35%

REDUCE BANDWIDTH CONSUMPTION ➔ 35%

- Require Oracle 11g ADVANCE COMPRESSION

- Can be enabled for all Redo Transport Methods and Protection Modes . 11g R1 can support ASYNC but check Metalink Note 729551.1

- SYNTAX

LOG_ARCHIVE_DEST_2='SERVICE=stnd by ASYNC COMPRESSION=ENABLE....'

or

EDIT DATABASE prim SET PROPERTY REDOCOMPRESSION=ENABLE;

HETEROGENOUS DATAGUARD SUPPORT

- Oracle 10g DATA GUARD supports mixed 32-bit and 64-bit configuration

- Oracle 11g DATA GUARD supports mixed WINDOWS / LINUX in same configuration

- Only PHYSICAL STANDBY
- Same ENDIANESS required on all Platforms

- Check Metalink Doc. 413484.1 for more support for Heterogenous Data Guard configuration

LOST WRITE DETECTION

- Faulty Storage Hardware / Firmware can lead to LOST WRITES to Data Corruptions.
- Very hard to diagnose such Data Corruptions when occurs
- This can cause OUTAGE in the Production Environment

11g Data Guard Detect Lost Writes

- Use DATA GUARD Physical Standby
- Compares versions of BLOCKS
  - Between Standby blocks and incoming Redo Stream
  - Version discrepancy can be on either Primary or Standby Database
- If Primary Database block Corruption is detected –> Resolved
  - Using Standby to Failover and
  - Restore Data Consistency
- New Initialization parameter - Less than 5% impact – Increase in Protection
  ALTER SYSTEM SET db_lost_write_protect= NONE|TYPICAL|FULL

AUTOMATIC BLOCK RECOVERY

- 11g R2 automatically repair corrupt data blocks using Active Data Guard copies of the corrupted blocks
- Automatic Block Media Recovery also automatically repair corrupted blocks discovered in the Physical Standby databases
- Reduce Production Downtime and avoid returning errors to your application
- Reduce Block Recovery time using up-to-date good blocks in real time instead of retrieving block from Disk, Tape or Flashback logs
- RMAN RECOVER BLOCK can be used for manual recovery searching several location for good copies of the data block.

```
RECOVER DATAFILE  3 BLOCK 3;

RECOVER
DATAFILE  6 BLOCK 3,9
DATAFILE  3 BLOCK 3;
```

- Physical Standby
- Flashback Logs
- Blocks in Full or Level 0 Backup

RECOVER BLOCK ..... EXCLUDE STANDBY

- Automatic Block Recover is applicable only for Physical Block Corruption like
  - When Checksum is Invalid
  - Block Contains all Zeros
  - Block Header is Fractured

ROLE BASED SERVICES

- Services only start based appropriate DATABASE ROLE
- Data Guard Broker interacts with Oracle Clusterware or Oracle Restart to make sure appropriate services are active after a ROLE TRANSITION
- Service is started when ROLE matches and MANAGEMENT POLICY is set to AUTOMATIC
- No Need to write database startup Triggers to make the service active in 11g R2.
- Must be created on both Primary and Standby clusters

srvctl add service -d prim -s TestServ -r node1,node2 -l PRIMARY -y AUTOMATIC
srvctl add service -d stdby -s TestServ -r node1,node2 -l PRIMARY -y AUTOMATIC

srvctl add service -d prim -s TestReport -r node1,node2 -l PHYSICAL_STANDBY
srvctl add service -d stdby -s TestReport -r node1,node2 -l PHYSICAL_STANDBY

```
srvctl add service -d <db_unique_name> -s <service_name>
[-l [PRIMARY][,PHYSICAL_STANDBY][,LOGICAL_STANDBY]
    [,SNAPSHOT_STANDBY]]
[-y {AUTOMATIC | MANUAL}]
```

➔MANAGEMENT POLICY

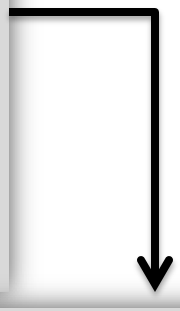
CONFIGURING ORACLE NET FOR ROLE BASED SERVICES

- FAILOVER=ON will enable Connect time failover
- Low value of CONNECT_TIMEOUT will minimize the delay if network address is unavailable
- Include both Primary and Standby Database SCAN/VIP based on the setup

```
sales =
    (DESCRIPTION=
       (FAILOVER=ON) <----
       (CONNECT_TIMEOUT=5) <----
       (ADDRESS_LIST=
          (ADDRESS=(HOST=boston)(PORT=1521))
          (ADDRESS=(HOST=dallas)(PORT=1521))) <----
       (CONNECT_DATA=(SERVICE_NAME=payroll))) <----
```

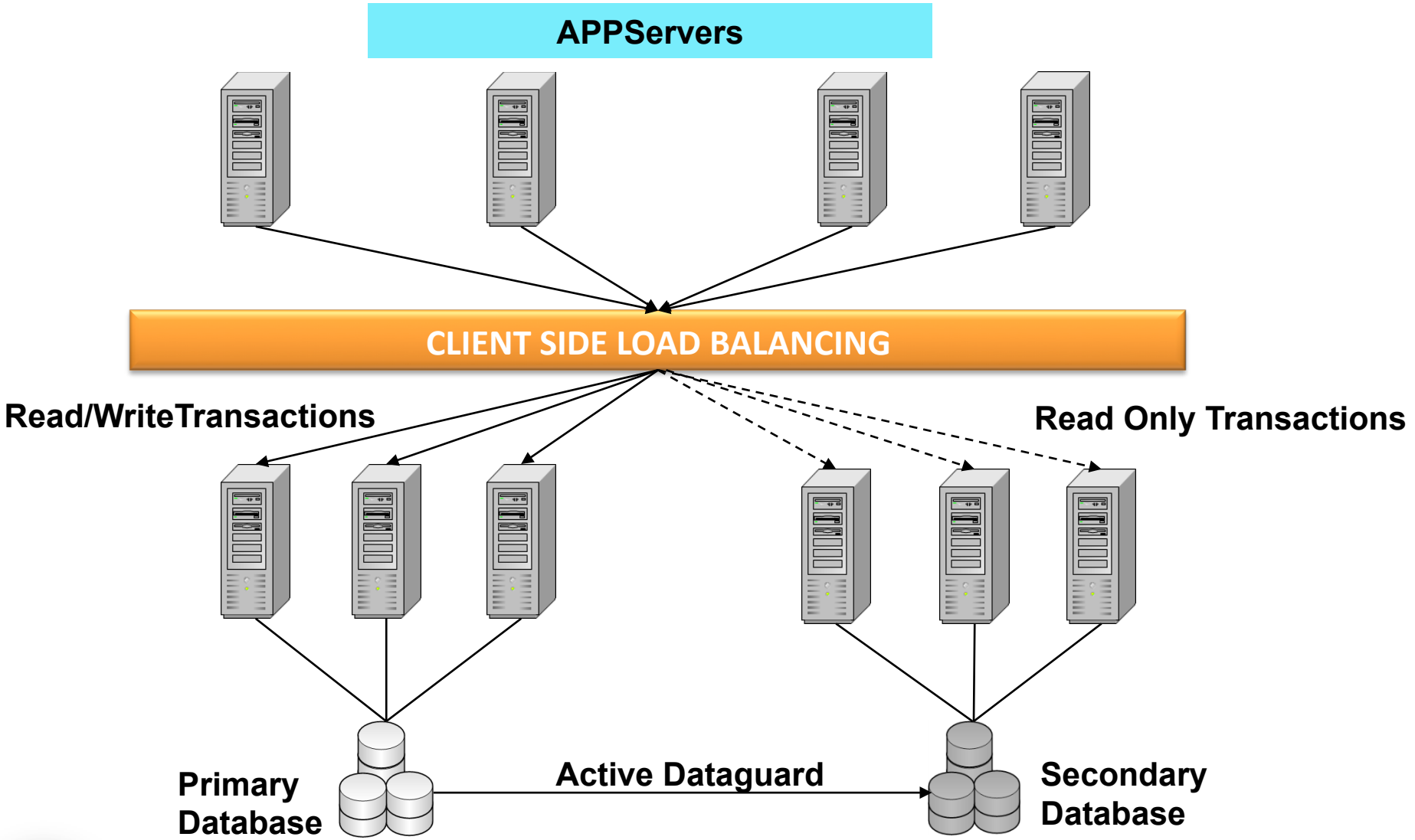
ROLE BASED SERVICES IN 11G R1

```
begin
dbms_service.create_service(
        service_name=>'DBFAILOVERSERV',
        network_name=>'DBFAILOVERSERV',
        clb_goal=>dbms_service.clb_goal_long,
        failover_method=>dbms_service.failover_method_basic,
        failover_type=>dbms_service.failover_type_select,
        failover_retries=>30,
        failover_delay=>5
        );
end;
/
```



```
CREATE OR REPLACE TRIGGER manage_siebel_service
after startup on database
DECLARE
        role VARCHAR(30);
BEGIN
        SELECT DATABASE_ROLE INTO role FROM V$DATABASE;
        IF role = 'PRIMARY' THEN
                DBMS_SERVICE.START_SERVICE('DBFAILOVERSERV');
        ELSE
                DBMS_SERVICE.STOP_SERVICE('DBFAILOVERSERV');
        END IF;
END;
/
```

TRANSACTION LOAD BALANCING



FLUSH REDO CLAUSE

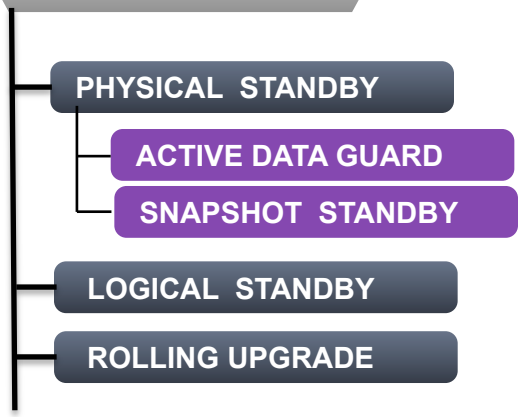
- Flush Redo data from Primary to Standby Database (Physical or Logical)
- Allow Failover to be performed with no Data Loss
- Issued on a MOUNTED but not OPEN Primary Database.

Issue the following SQL statement at the primary database:

```
SQL> ALTER SYSTEM FLUSH REDO TO target_db_name;
```

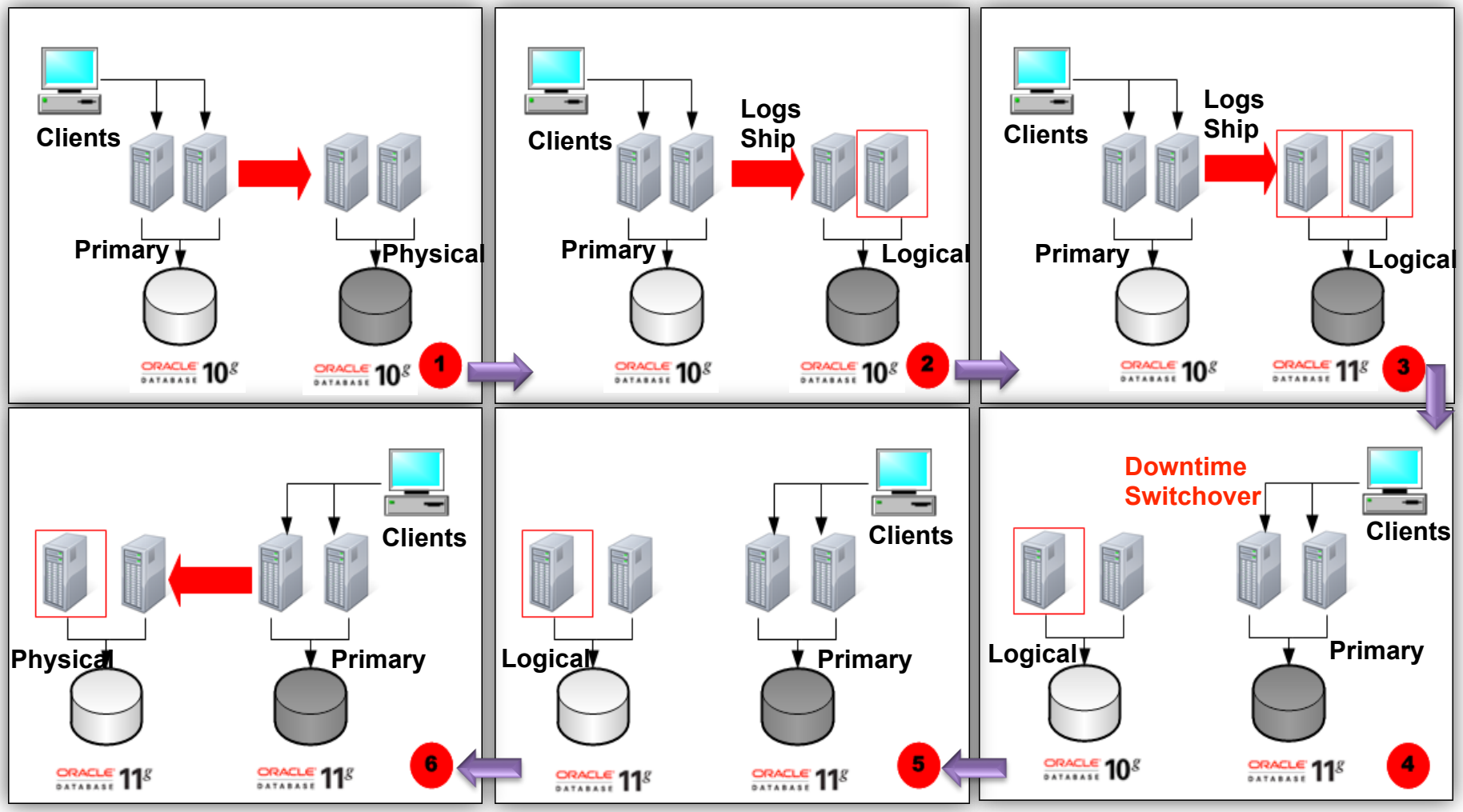
For `target_db_name`, specify the `DB_UNIQUE_NAME` of the standby database that is to receive the redo flushed from the primary database.

11g DATAGUARD



11g DATAGUARD
ROLLING UPGRADE

ROLLING UPGRADE USING TRANSIENT LOGICAL STANDBY



QUESTIONS & ANSWERS

indy.johal@datasoftech.com

THANKS

indy.johal@datasoftech.com