Oracle Data Guard for High Availability and Disaster Recovery

John Zhong

Senior Technical Specialist
Global Database Management Group
Citigroup
and
Vice President of
American DBAOnline

John_j_zhong@yahoo.com

Agenda

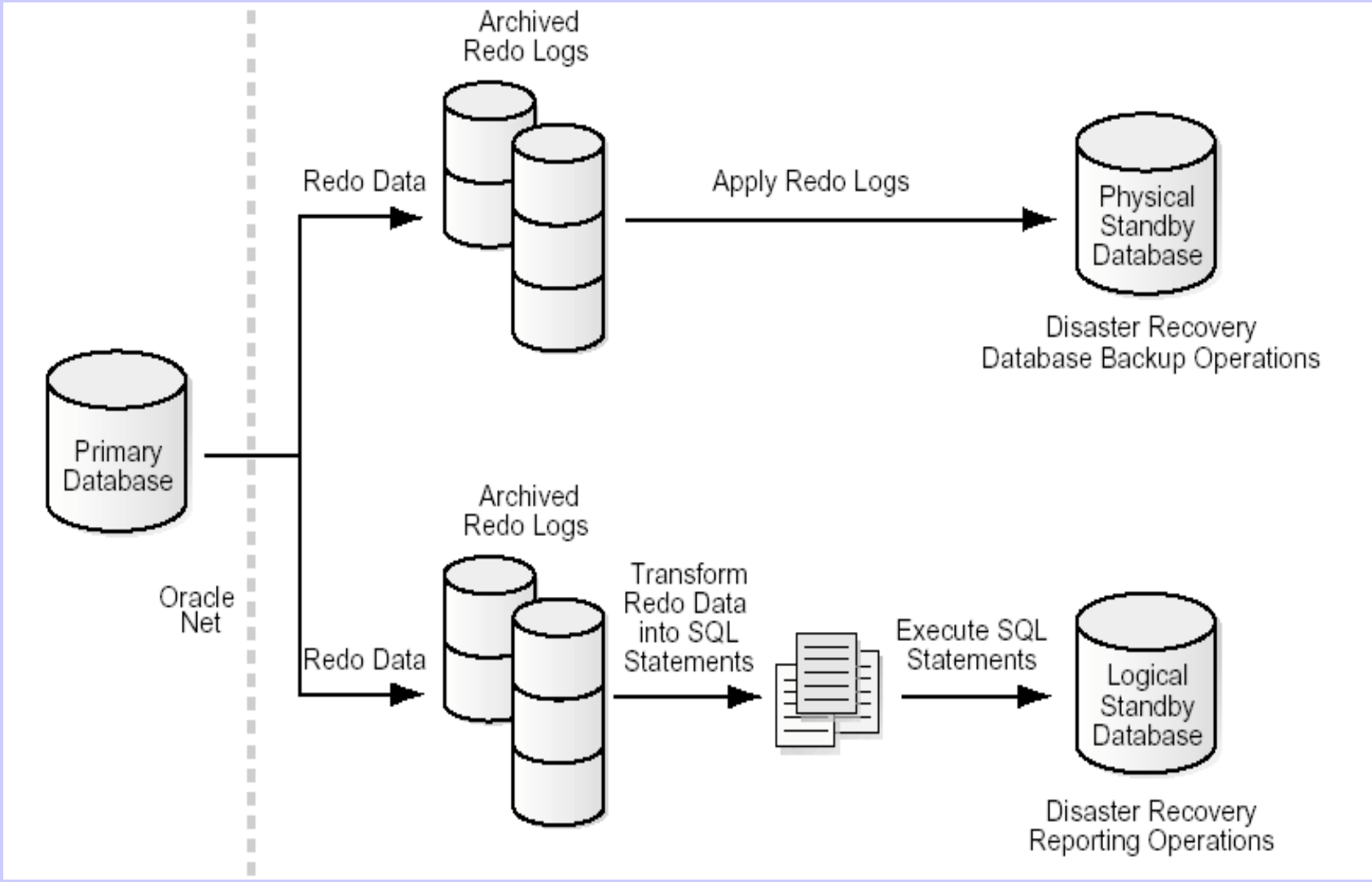
- Data Guard Overview
- Data Guard 10g New Features
- Data Guard Configuration
- Automatic Gap Detection and Resolution
- Redo Shipping with LGWR
- Data Protection Modes
- Managing Data Guard
- Monitoring Data Guard
- High Availability and Disaster Recovery

Data Guard Overview

- Data Guard Configurations
  - Primary Database
  - Standby Databases
- Data Guard Services
  - Log Transport
  - Log Apply
  - Role Management
- Data Protection Modes
- Data Guard Broker

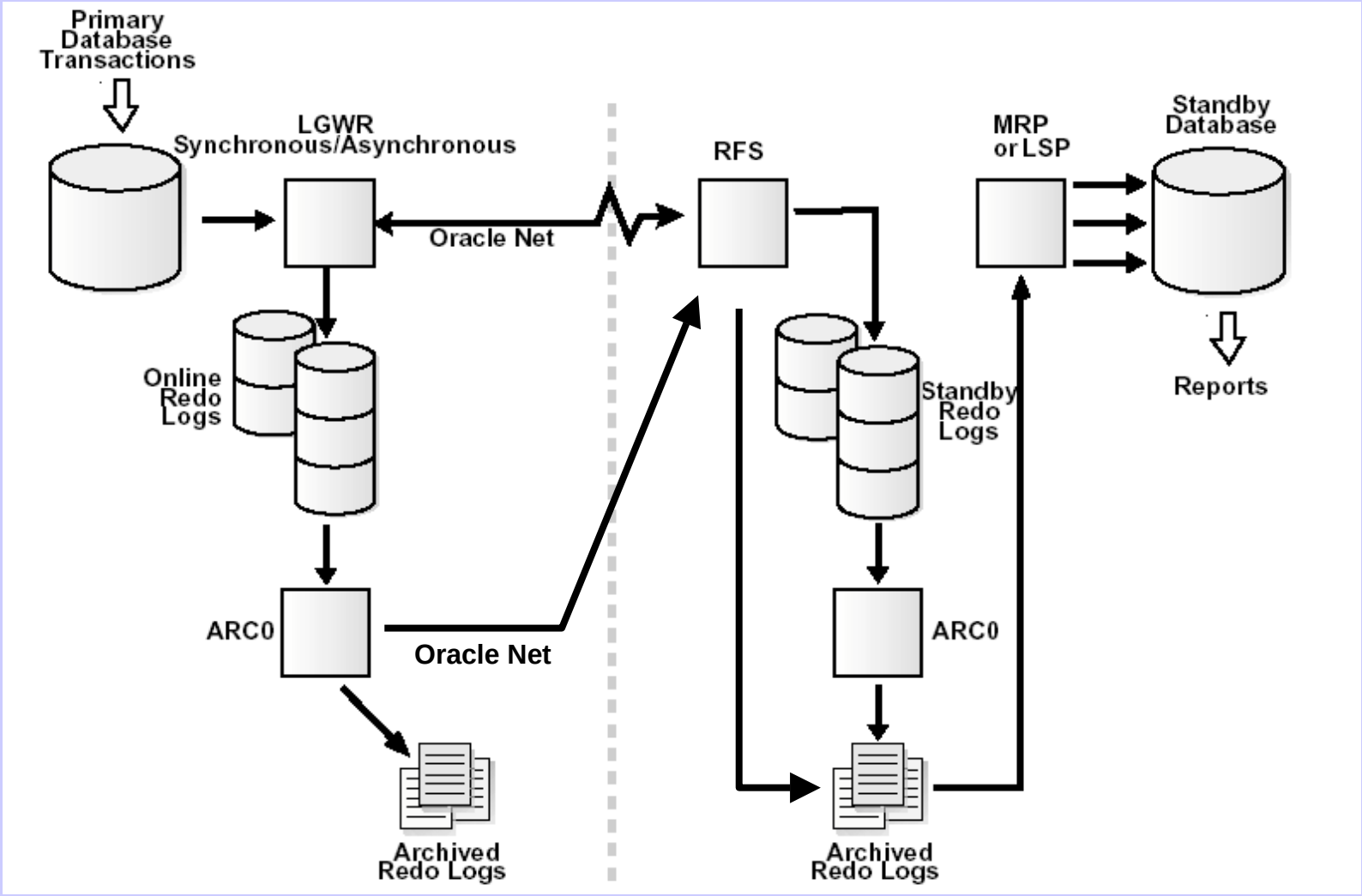
Data Guard Overview

Data Guard Configuration



Data Guard Overview

Data Guard Configuration: Physical Standby Database



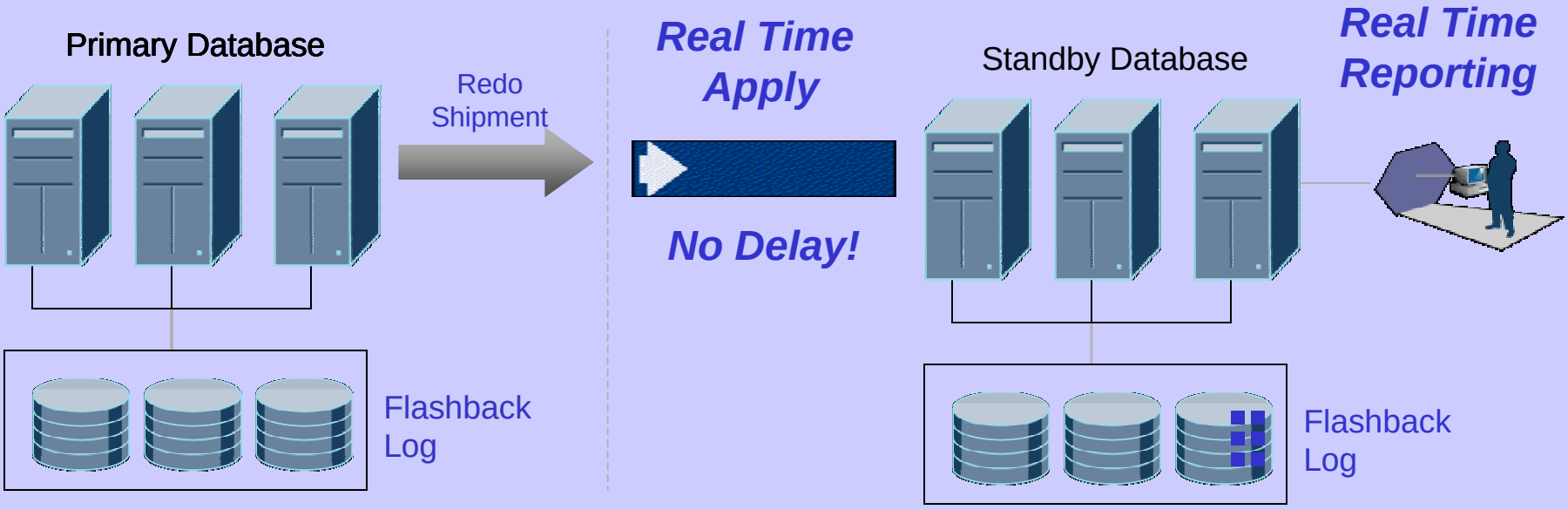
Data Guard Features

- Automatic log shipping
- Automatic redo gap detection and resolution
- Redo shipping using LGWR
- Data protection modes
- Role transition management
  - Failover and Switchover

Data Guard 10g New Features

- Real Time Apply
- Flashback Database Integration
  - Flashback standby after OPEN RESETLOGS on primary, and remove the need to recreate standby.
  - Flashback failed primary after failover and convert it to a standby.
- New Default Behavior for the STARTUP, MOUNT, and OPEN Statements on Physical Standby
- RAC Integration
- Simplified Browser-based Interface Focused on Best Practices

Data Guard 10g New Features



Primary: No reinstantiation after failover!

Data Guard Configuration

- Creating a physical standby database
  - Tasks on primary
  - Tasks on standby

Data Guard Configuration – Tasks on Primary

- Enable archiving:
  - Database in ARCHIVELOG mode
  - Set local archive destination: LOG_ARCHIVE_DEST_1
  - Enable automatic archiving:

    log_archive_start = true

    or alter system archive log start;

Data Guard Configuration – Tasks on Primary

- Backup primary database
- Create standby controlfile, for example,

  alter database create standby controlfile as 'TEST1_stdby.ctl';

- Prepare init file:

  log_archive_start = true
  log_archive_dest_1 = 'LOCATION=<arch_directory> MANDATORY REOPEN=30'
  log_archive_dest_state_1=enable
  log_archive_format = "log_%s_%t.arc"

  remote_archive_enable=true
  log_archive_dest_2 = 'service=<standby_tns> reopen=30‘
  log_archive_dest_state_2=enable

- Add standby_tns to tnsnames.ora
- Copy primary database backup and standby controlfile to standby
- Start up listener

Data Guard Configuration – Tasks on Standby

- Set environment for standby database
  - ORACLE_SID for standby does not have to be the same as primary
- Create data file and admin directories:

  /oradata/<db_name>

  $ORACLE_BASE/admin/<db_name>/udump, bdump, cdump, arch
- Prepare init file:

```
log_archive_start = true
log_archive_dest_1 = 'LOCATION=<arch_directory> MANDATORY
   REOPEN=30'
log_archive_dest_state_1=enable
log_archive_format = "log_%s_%t.arc"
remote_archive_enable=true

# Standby role parameters
standby_archive_dest=<standby_arch_dir>
standby_file_management=auto
fal_server=<primary_tns>
fal_client=<standby_tns>
```

Data Guard Configuration – Tasks on Standby

- Add TNS entries primary_tns and standby_tns to tnsnames.ora
- Restore database backup:
  - Restore database files
  - Restore archive logs
  - Copy standby controlfile to directories as specified in init file
- Startup and mount standby database:

  startup nomount

  alter database mount standby database;
- Recover managed standby database:

  alter database recover managed standby database disconnect from session;

Data Guard Configuration – Tasks on Standby

- Open database in read only mode:

  alter database recover managed standby database cancel;

  alter database open read only;

- Add temp file to locally managed temporary tablespace:

  alter tablespace temp add tempfile 'file_name' size <nnn> reuse;

- Switch back to managed recover mode:

  alter database recover managed standby database disconnect from session;

- Startup standby listener
- Standby is created and receiving / applying logs

Data Guard Configuration – Verify Standby

- Check log shipping:
  - Run this query on primary and standby:

    select sequence# from v$archived_log order by 1;
  - Do a log switch from primary, run the query again, the log should be shipped to standby automatically
- Check log apply:
  - Put standby in managed recover mode.
  - Run query to get last applied log:

    select SEQUENCE#, APPLIED from v$archived_log order by 1;
  - Do a log switch from primary, new archived logs should be applied.
- Check database changes are synced:
  - Make some changes on primary and do a log switch
  - Make sure last archived log is shipped and applied on standby
  - Open standby in read only, and verify the changes.

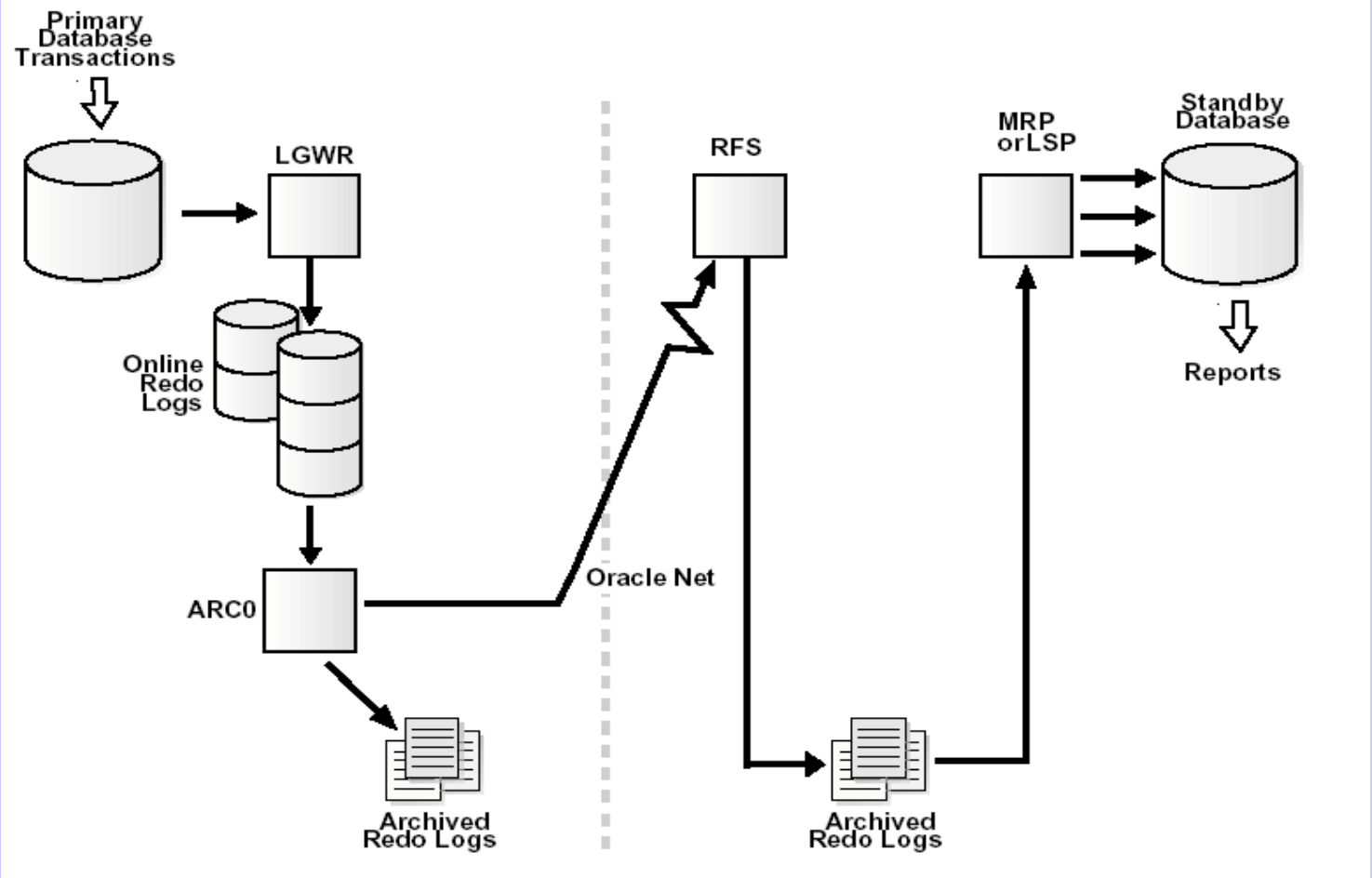
Automatic Gap Detection and Resolution

- Automatic Gap Resolution
  - ARCH (primary) sends log to standby
  - RFS (remote file server process on standby) receives the redo and detects gap
  - ARCH sends missing logs as requested by RFS
  - Starting 9.2.x, ARCH on primary polls standby every minute to see if any gap and sends missing logs
- Gap Resolution by Fetch Archive Log (FAL)
  - Managed recover process (MRP) applies log on standby.
  - If MRP finds missing or corrupted logs, it asks primary to fetch archive log (FAL).
  - Two parameters are needed for FAL to work

    fal_server=<primary_tns>

    fal_client=<standby_tns>

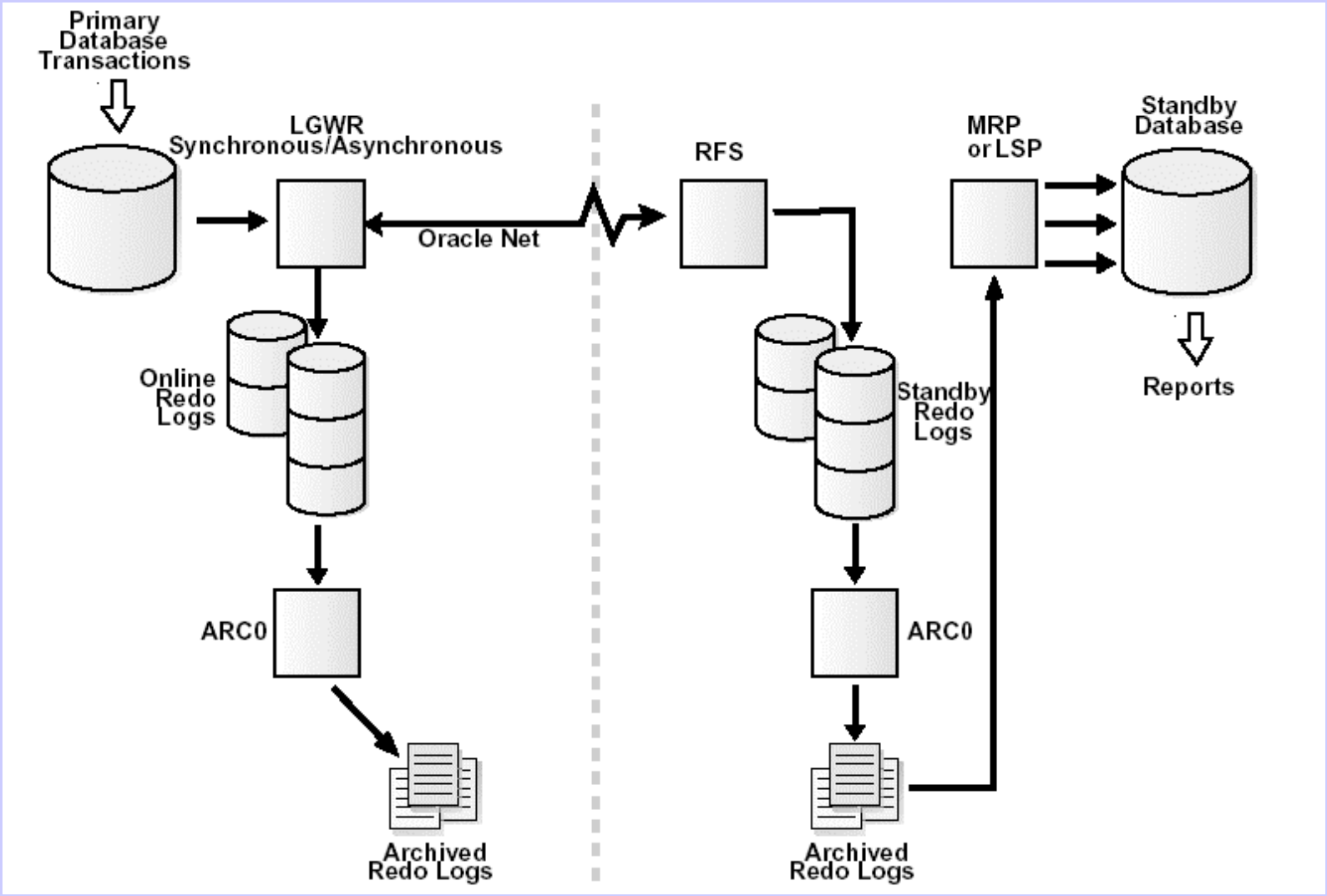
Automatic Gap Detection and Resolution



Redo Shipping with LGWR

- Redo data can be transmitted to standby by
  - ARCH process or
  - LGWR (log writer) – Oracle Data Guard feature. Redo log is shipped directly to standby as it is being written to local online redo log.
- Redo shipping with LGWR
  - SYNC
    - Zero data loss
    - Performance impact
  - ASYNC
    - Minimum data loss
    - Little performance impact

Redo Shipping with LGWR



Redo Shipping with LGWR

- Redo shipping with LGWR:
  - Create standby redo logs on standby. Create same number of standby redo logs as primary's:

```
alter database add standby LOGFILE
('/oradata/TEST1/s_redo_01a.log',
 '/oradata/TEST1/s_redo_01b.log')
) SIZE 100M, …;
```

  - Configure primary parameter

log_archive_dest_2 = 'service=<standby_tns> **LGWR SYNC|ASYNC** reopen=30'

Data Protection Modes

- Maximum protection
  - Redo transmission: LGWR SYNC
  - Zero data loss. Transaction will not commit until redo is written to standby
  - Primary will shutdown if it can not write redo to a standby
  - Impact on the performance and availability of the primary database.
- Maximum availability
  - Redo transmission: LGWR SYNC
  - Zero data loss. Transaction will not commit until redo is written to standby
  - Primary will NOT shutdown if it can not write redo to standby (Lower to maximum performance mode).
  - Impact on the performance, but not on availability of primary database

Data Protection Modes

- Maximum performance
  - Default mode
  - Redo transmission: LGWR ASYNC or ARCH (default)
  - Transaction will not wait to commit until redo is written to standby.
  - Minimum data loss, little impact on the performance and no impact on availability of the primary database.
  - Data loss exposure is limited by the size of the configurable ASYNC buffer

    LGWR ASYNC=<buffer size>

    Maximum buffer size: 10MB (9i), 50MB (10g)
- Set database protection mode:

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

Managing Data Guard

- Startup and shutdown standby database
- Recover managed standby database
- Open in read only mode
- Create primary backup from standby
- Recover when standby data files are corrupted.
- Resolve log gaps
- Manage database files
- Switchover
- Failover

Managing Data Guard

- Manage database files
  - standby_file_management=auto
  - New datafiles created on the primary database are automatically created on the standby
  - Renaming datafiles in the primary database is not propagated to the standby database. Need to rename the same datafiles on the standby.

Managing Data Guard

Graceful switchover – planned maintenance, DR test

- Prepare switchover
  - Edit / verify init.ora on primary and standby to support role transition
  - Verify that there are no active users connected to either database.
  - Do a log switch on primary. Recover standby until the last archived log applied
- Convert primary database to new standby (perform this on primary)
  - alter database commit to switchover to physical standby with session shutdown;
  - Shutdown immediate
  - edit parameters in init.ora for standby role
  - startup nomount and mount new standby

Managing Data Guard

Graceful switchover

- Convert standby database to new primary (perform this on standby)
  - alter database commit to switchover to primary;
  - shutdown immediate
  - Edit parameters in init.ora for primary role
  - startup

Managing Data Guard

Failover

- Finish recovery of standby database

  LGWR log shipping:

  alter database recover managed standby database finish;

  ARCH log shipping

  alter database recover managed standby database finish skip standby logfile;

- Convert the standby database to the primary role

  alter database commit to switchover to primary;

- Shutdown immediate
- Edit init.ora for primary role
- Startup

Monitoring Data Guard

- Alert log
- V$ views

Monitoring Data Guard

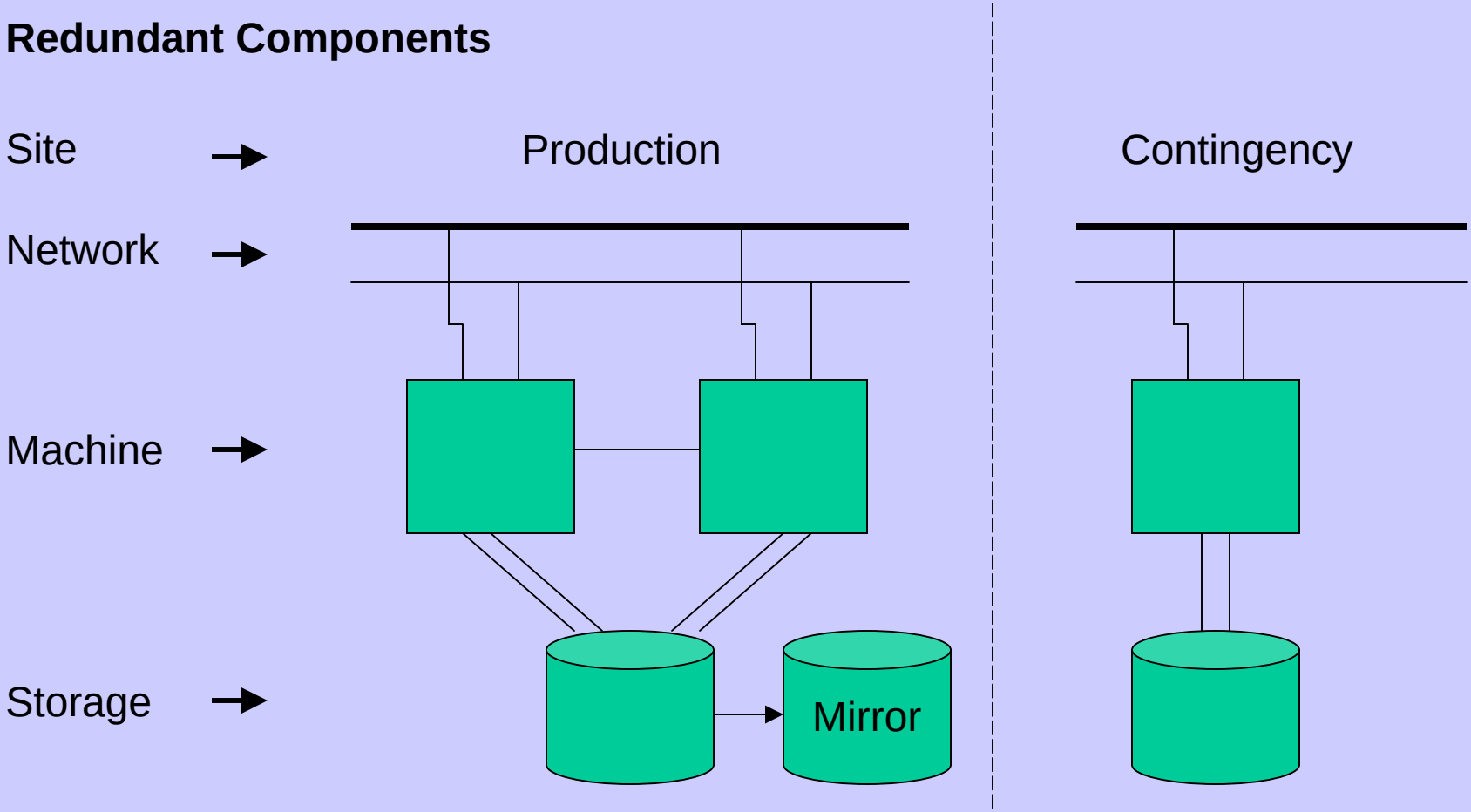
- Primary and standby instances
- Primary and standby listeners
- Standby mode
  - not mounted
  - mounted
  - read only
  - managed recovery

Monitoring Data Guard

- Redo shipping
  - Alert log or v$dataguard_status
  - v$ managed_standby (standby) or v$archive_dest (primary)
- Standby processes: v$managed_standby
  - ARC0
  - MRP/MRP0
  - RFS
- Archive log latency
  - v$archived_log (primary, standby)
- Archive log gap
  - v$archived_log (primary or standby), v$archive_gap (standby)
- Apply log latency
  - v$log_history, v$archived_log (standby, primary)

High Availability and Disaster Recovery

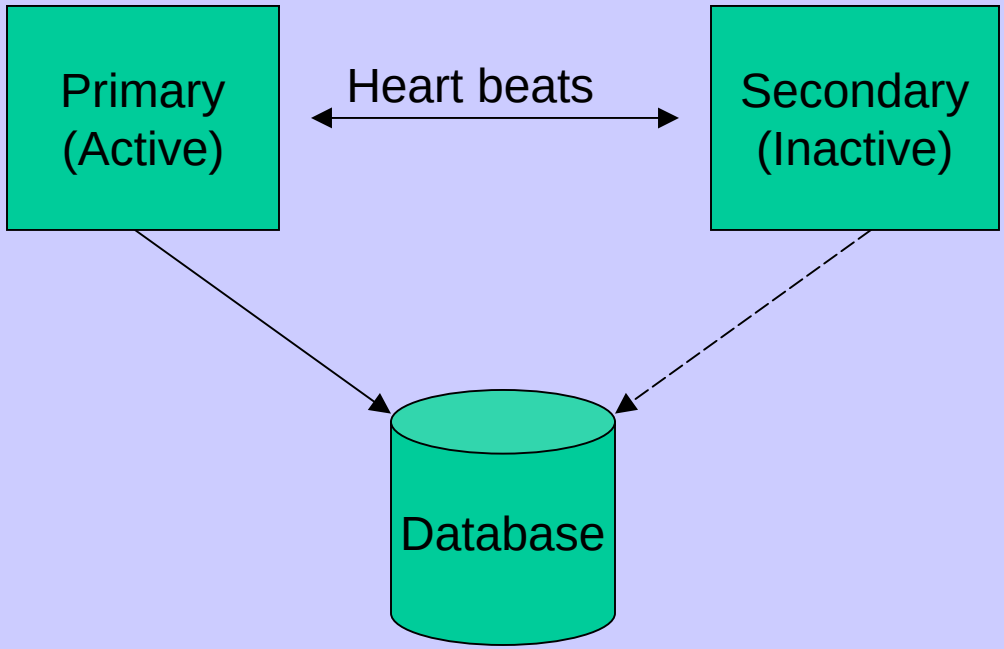
Redundant Components



High Availability

Server Cluster – Active / Inactive

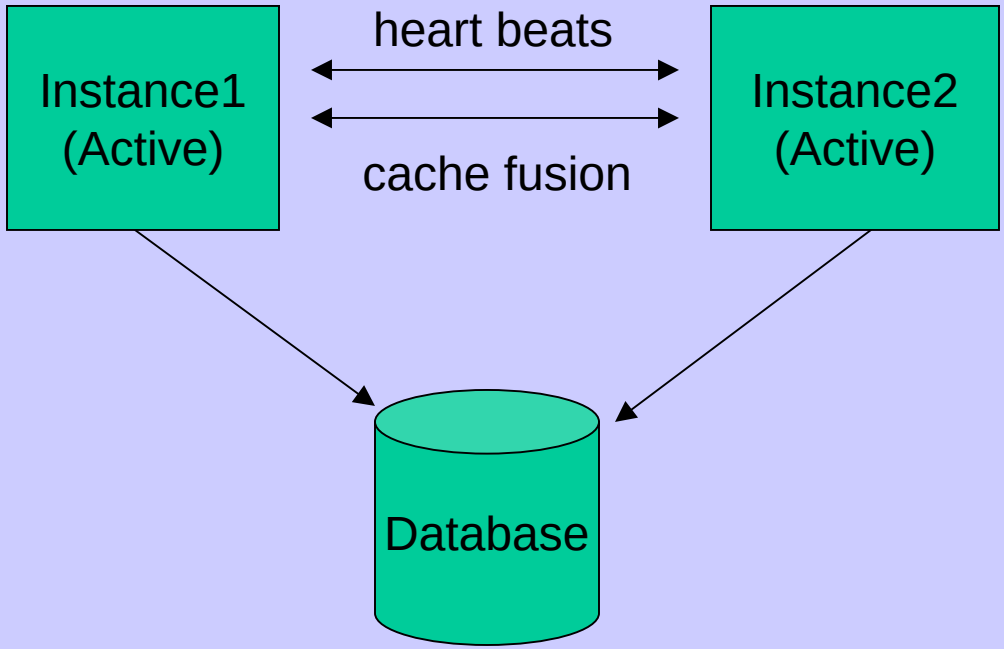
- Protect from machine or instance failure
- Some down time to failover if primary fails.
- Not protect from storage failure or disasters



High Availability

Real Application Cluster

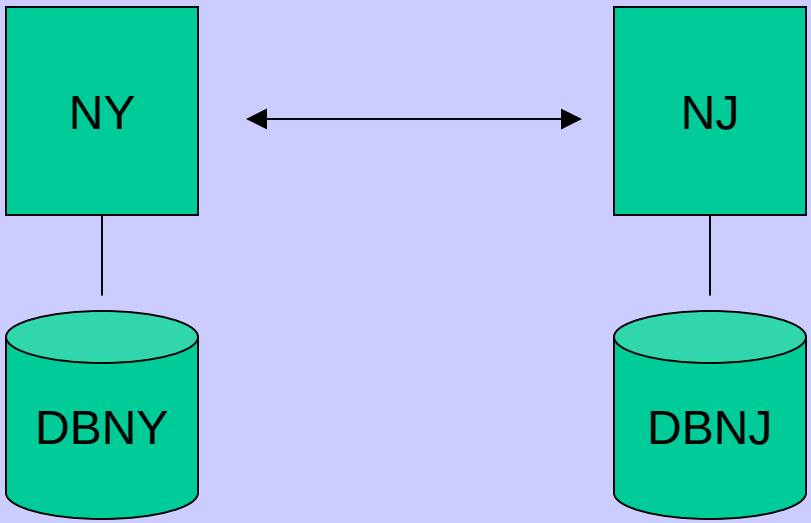
- Protect from machine or instance failure
- No down time if a cluster node or an instance fails.
- Not protect from storage failure or disasters



High Availability and Disaster Recovery

Replications:

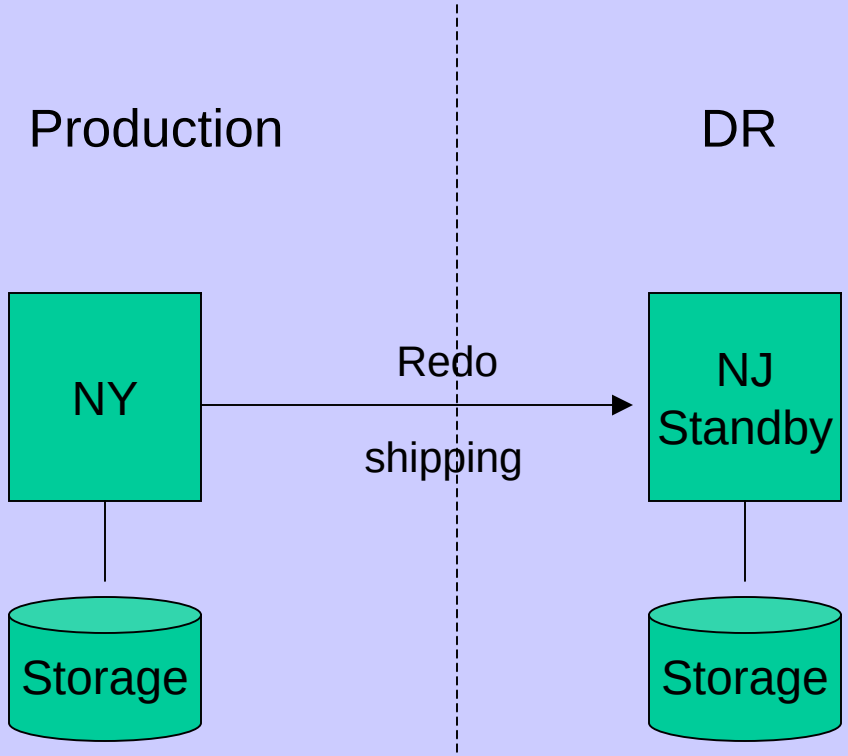
- Protect from machine and storage failure or disasters
- Database changes on both sites, hot-hot configuration
- Issues with DBA maintenance cost and a small window of down time and data loss



High Availability and Disaster Recovery

Data Guard

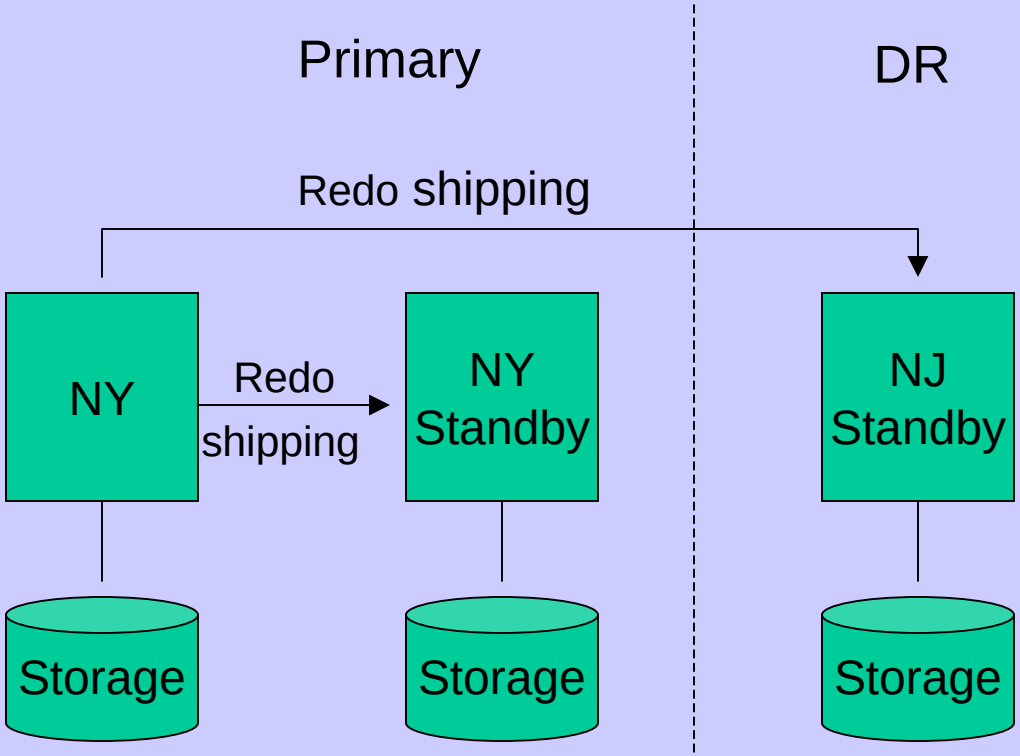
- Protect from machine and storage failure or disasters
- Protect data loss
- Easy to setup and maintain
- Some down time to failover if primary fails.



High Availability and Disaster Recovery

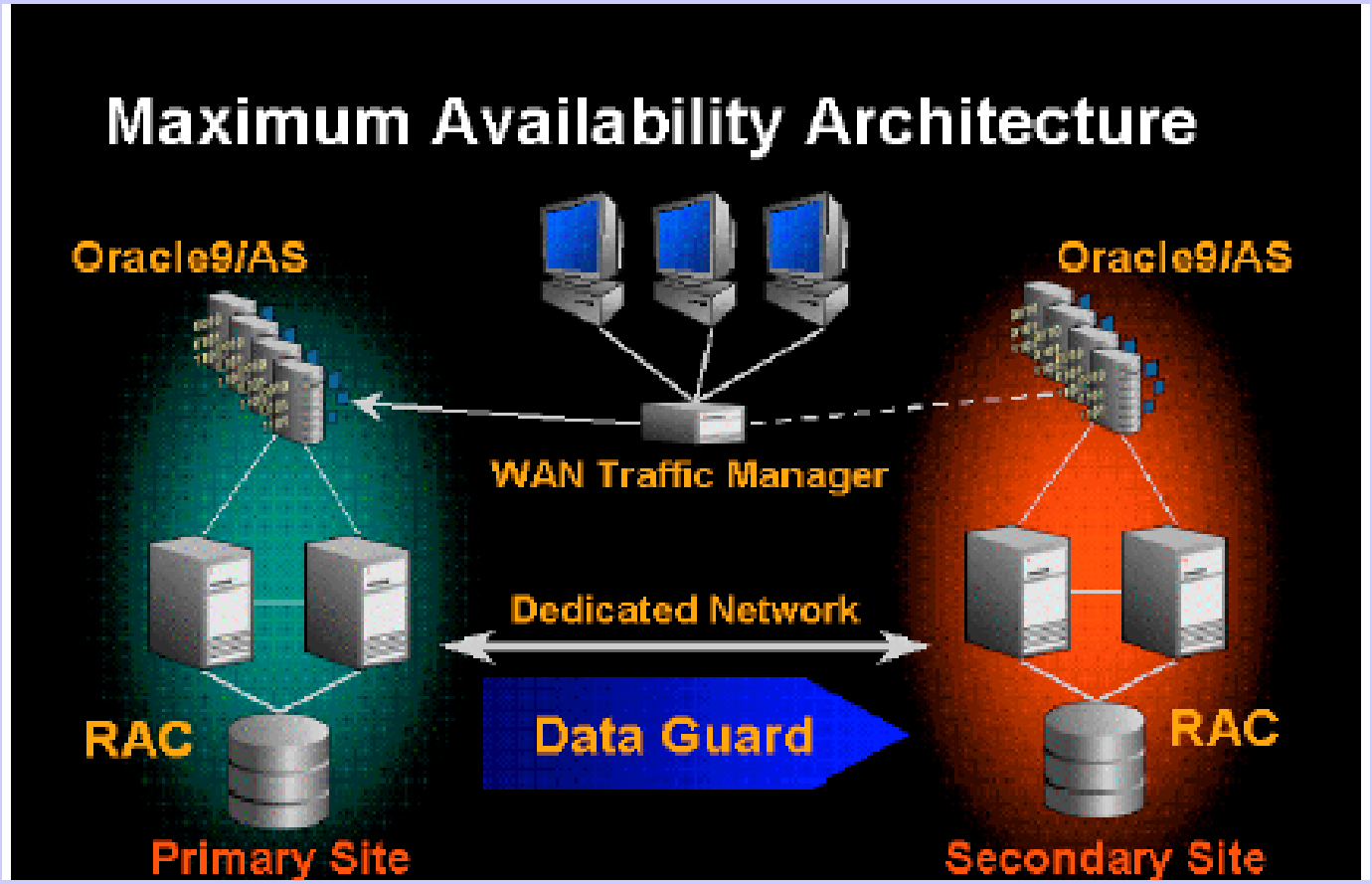
Data Guard

- Data Guard can be used for HA and disaster recovery



High Availability and Disaster Recovery

Maximum Availability Architecture (MAA): combined RAC and Data Guard



Summary

- Automatic gap detection and resolution
- Redo shipping with LGWR
- Data protection modes
- High availability
- Disaster recovery
- Simple management
- Efficient use of system resource

Q & A