Oracle Data Guard 11*g*
The Next Era in Data Protection and Availability

An Oracle Technical White Paper
August 2007

ORACLE

Oracle Data Guard 11g
The Next Era in Data Protection and Availability

Oracle Data Guard 11g
The Next Era in Data Protection and Availability

Data Guard 11g Highlights

Physical standby database open read/write for test or other purposes with zero compromise in data protection

ASYNC transport enhanced to eliminate the impact of latency on network throughput

Automatic failover for Maximum Performance (ASYNC)

Automatic failover configurable for immediate response to designated events or errors

Fast detection of corruptions caused by lost writes in the storage layer

More flexibility in primary/standby configurations (e.g. Windows primary and Linux standby)

SQL Apply supports XML data type (CLOB)

Many performance, manageability, and security enhancements

Support for new Oracle Database 11g Options – Oracle Active Data Guard and Oracle Advanced Compression

INTRODUCTION

Efficient business operations, high quality customer service, conformance with government regulations, and safeguarding corporate information assets all depend upon achieving the highest possible level of data protection and availability. Thus it is no surprise that data protection and availability are among the top priorities of business continuity initiatives for companies of all sizes and industries.

On the surface, products and services to address these requirements appear to be mature and unexciting. Backup and recovery from tape, storage based remote-mirroring, and database log shipping are the traditional solutions for data protection and disaster recovery (DR) requirements. Unfortunately, traditional solutions also have the traditional shortcomings of being unable to reliably deliver aggressive objectives for both recovery point (data protection) and recovery time (high availability), or do so in a way that delivers maximum return on investment by avoiding underutilized assets, high acquisition and support costs, and increased complexity.

In contrast, Oracle Data Guard 11*g* [1] redefines what users should expect from a disaster recovery solution. It can address both High Availability and Disaster Recovery requirements, and is the ideal complement to Oracle Real Application Clusters (Oracle RAC). Data Guard has the requisite knowledge of the Oracle database to reliably protect a standby database from corruptions that attempt to propagate from a primary database. It is straightforward to implement and manage. It also enables all standby databases, both physical and logical, to be used for productive purposes while in standby role. Data Guard delivers:

- Reliability– optimum data protection and availability. You always know the state of your standby database and it can very quickly (in seconds), assume the primary role.
- Lower cost and complexity – mature capabilities and rich management interface, with most features included in Oracle Enterprise Edition
- Maximum return on investment – All standby databases can be utilized for production purposes while in standby role. Idle resources are eliminated WITHOUT increasing complexity.

Regardless of the degree to which high-availability has previously been built into your systems and infrastructure, data protection and availability are universally enhanced by including Data Guard in your IT architecture.

This white paper provides an architectural and technology overview of Data Guard 11*g*. For additional details, please refer to Oracle Data Guard documentation [2].

ORACLE DATA GUARD - OVERVIEW

"Fidelity National Financial successfully implemented Oracle RAC and Grid infrastructures. Additionally, Oracle Recovery Manager was deployed to implement a Backup and Recovery capability and Data Guard was utilized to provide remote disaster recovery and High Availability across a Wide Area Network. Collectively, the utilization of Oracle High Availability Features and implementation utilizing Oracle Maximum Availability Architecture (MAA) best practices has enabled FNF to meet service level agreements at the lowest cost."

- Charles Kim
Chief Oracle Database Engineering Counsel
Fidelity Information Services

Data Guard is a central component of an integrated Oracle Database High Availability (HA) solution set that helps organizations ensure business continuity by minimizing the various kinds of planned and unplanned downtime that can affect their businesses. The following diagram shows the various HA features available with Oracle Database 11*g*. For further details on each of these features, please refer to Oracle Database High Availability [3], on the Oracle Technology Network.

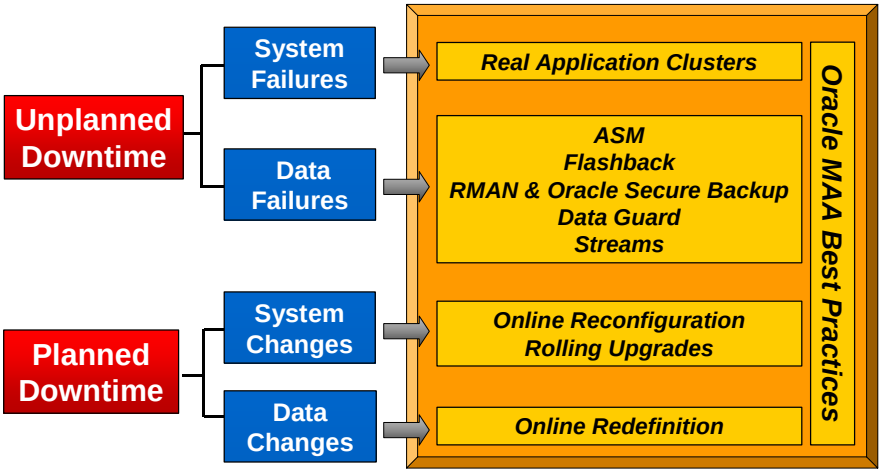

Figure 1 – Integrated High Availability Features of Oracle Database 11g

Data Guard provides the management, monitoring, and automation software infrastructure to create, maintain, and monitor one or more standby databases to protect enterprise data from failures, disasters, errors, and data corruptions. If the user desires, Data Guard will automatically failover production to a standby system if the primary fails in order to maintain high availability required for mission critical applications. In addition to providing HA/DR, Data Guard standby databases can also support production functions for reporting, query, backup and test, while in a standby role.

Data Guard is comprised of time-proven services in three different areas: Redo Transport Services, Apply Services, and Role Management Services.

Data Guard Redo Transport Services: A Data Guard configuration includes a production database, also known as the primary database, and up to nine standby databases. Data Guard maintains synchronization of primary and standby databases using redo data. As transactions occur in the primary database, redo data (the information required to recover a transaction) is generated and written to the local redo log files. *Data Guard Redo Transport Services* are used to transfer this redo data to the standby site synchronously or asynchronously. This flexibility enables standby databases to be located at remote disaster recovery sites thousands of miles away from the production data center, or they may be located in the same city, same campus, same building, or even the same computer room.

Data Guard Apply Services: Once redo data has been transferred to the standby site, *Data Guard Apply Services* provide two different methods for applying data to a standby database: Data Guard Redo Apply (physical standby), or Data Guard SQL Apply (logical standby). A physical standby database has on-disk database structures that are identical to the primary database on a block-for-block basis, and is updated using Oracle media recovery. A logical standby database is an independent database that contains the same data as the primary database and is updated using SQL statements. Each method has different advantages giving customers the flexibility of selecting the method that is the best choice for their requirements. These differences are discussed in detail later in this paper.

"Data Guard Fast-Start Failover provides simple, fast, unattended failover for our outage management system that PPL depends upon to provide critical customer services 24 hours a day and especially during emergencies. While we have used Data Guard for disaster recovery (DR) since Oracle9i, Fast-Start Failover blurs the line between High Availability and DR – enabling us to address both requirements with a single solution"

- Chris Carter, Director, Enterprise Technology Services PPL Services Corp.

Data Guard Role Management Services: If the production database becomes unavailable because of a planned or an unplanned outage, *Data Guard Role Management Services* quickly switch a chosen standby database to the production role, minimizing the downtime and preventing data loss. Data Guard offers the flexibility of executing role transitions via manual control, or automatically without operator intervention according to a set of rules that are user-configurable. Data Guard provides the infrastructure to automate client failover to provide the level of high availability required for mission critical applications. After failover, the failed primary can be rapidly reinstated as a standby database of the new primary, quickly returning the configuration to a protected state.

Creating and Managing a Data Guard Configuration: The primary and standby databases, as well as their various interactions, may be managed by using SQL*Plus. Data Guard also offers a distributed management framework called the Data Guard Broker, which automates and centralizes the creation, maintenance, and monitoring of a Data Guard configuration. Administrators may interact with the Broker using either Enterprise Manager Grid Control or the Broker's command-line interface (`DGMGRL`).

Utilization of Standby Resources: All Data Guard standby databases can enable up-to date read access to the standby database while redo being received from the primary database is applied. This makes all standby databases excellent

candidates for offloading the primary database of the overhead of supporting read-only queries and reporting.

All Data Guard standby databases support online backups using RMAN [4]. Because a physical standby database is an exact replica of the primary database, a physical standby database can be used to offload the primary of the overhead of performing backups.

A physical standby database (Redo Apply) can be converted into a Snapshot Standby to enable hot patch and other testing using a true read-write replica of production data. Data Guard maintains data protection at all times by continuing to ship redo data to the Snapshot Standby where it is archived for safe-keeping and available to rapidly resynchronize when it is converted back to being a standby database after testing is complete.

A logical standby (SQL Apply) database has the additional flexibility of being open read-write. While data that is being maintained by SQL Apply cannot be changed, additional local tables can be added to the database, and local index structures can be created to optimize reporting or to utilize the standby database as a data warehouse or to process information used to load data marts.

"Data Guard Rolling Upgrade resulted in limited downtime during a complete upgrade of our database (Oracle10.1.0.3 to 10.1.0.4), servers, and storage. We had no downtime for read access, and less than 30 minutes of downtime for read/write transactions"

- Dan Dressel, Database Architect Thomson Legal & Regulatory (OpenWorld 2006)

A logical standby database can be used to perform rolling database upgrades, minimizing downtime when upgrading to new database patchsets or full database releases. Furthermore, a physical standby can also be temporarily converted to a *transient* logical standby to execute a rolling database upgrade, and then returned to its normal operating state as a physical standby.

DATA GUARD 11*g*

Data Guard 11*g* includes new features to extend its already mature capabilities for data protection and disaster recovery, and includes enhancements to high availability capabilities first introduced in Data Guard 10g Release 2. It provides new ways to make productive use of a physical standby database while in standby role, without compromising data protection. Data Guard 11g also supports new Oracle Database 11*g* Options that leverage Data Guard technology to generate additional benefits for Oracle customers. A summary of these enhancements is provided in the following sections.

New capabilities included with Data Guard 11*g*

Snapshot Standby: This is a new type of standby database that is created from a physical standby database. Once created, a snapshot standby can be opened read-write to process transactions that are independent of the primary database for test or other purposes. A snapshot standby database will continue to receive and archive updates from the primary database, however, redo data received from the primary will not be applied until the snapshot standby is converted back into a

physical standby database and all updates that were made while it was a snapshot standby are discarded. This enables production data to remain in a protected state at all times.

Automatic failover for SYNC and ASYNC redo transport: Data Guard 10*g* Release 2 introduced automatic failover using the new feature Fast-Start Failover with Maximum Availability protection mode (SYNC). Data Guard 11*g* extends Fast-Start Failover to also support Maximum Performance mode (ASYNC) by adding a user configurable data loss threshold that guarantees an automatic failover will never result in data loss that exceeds the desired recovery point objective (RPO).

Users can configure an automatic failover to occur immediately, without waiting for the Fast-Start Failover threshold time period to expire based on designated health check conditions or any desired ORA-nnnnn error.

A new `DBMS_DG` PL/SQL packaged enables applications to notify the Fast-Start Failover Observer process to initiate an automatic failover.

"Data Guard is our preferred Disaster Recovery solution for Oracle data. We have very high performance applications. No other 3rd party product has approached the throughput that Data Guard can attain in zero data loss configurations."

- Manohar Malayanur
Manager and Infrastructure Architect
Fannie Mae

Performance Enhancements: Many performance enhancements include:

- Parallel media recovery (physical standby) significantly enhances physical standby apply performance for all workload profiles.
- SQL Apply enhancements (logical standby) will increase apply performance for inserts and updates to tables that are not partitioned and that do not contain LOB, LONG or XML type column. Also, SQL Apply now applies parallel DDL in parallel, rather than serially as was the practice in previous releases.
- ASYNC redo transport enhancements increase network throughput by eliminating the impact of network latency, particularly significant in WAN deployments.
- Further reduction in failover times when using Fast-Start Failover

Transient Logical Standby: Users can convert a physical standby to a *transient* logical standby database to effect a rolling database upgrade, and then revert to a physical standby once the upgrade is complete (using the `KEEP IDENTITY` clause). This benefits physical standby users who wish to execute a rolling database upgrade without investing in redundant storage otherwise needed to create a logical standby database.

Enhanced Data Protection: A Physical Standby can now detect lost datafile writes caused by faulty storage hardware and firmware that lead to data corruption. Data Guard will compare versions of blocks on the standby with that of the incoming redo stream. If there is a version discrepancy it implies a lost write. The user can then failover to the standby database and restore data consistency.

"Data Guard Logical Standby is an important component of a long-term strategic hardware and software platform, dramatically increasing capacity and scalability for our users. After implementing this complete solution, we achieved performance improvements of 50-95% in most batch processing operations."

- David Sink,
Business Intelligence Architect
e-Rewards Market Research

Enhanced Security: SSL authentication can be used in lieu of password file to authenticate redo transmission. Note: SSL authentication requires use of PKI Certificates, ASO and OID.

Additional SQL Apply Data Type Support: SQL Apply continues to add support for additional data types, other Oracle features, and PL/SQL, including:

- XMLType data type (when stored as CLOB)
- Ability to execute DDL in parallel on a logical standby database
- Transparent Data Encryption (TDE)
- `DBMS_FGA` (Fine Grained Auditing)
- `DBMS_RLS` (Virtual Private Database)

SQL Apply Manageability Enhancements: Scheduler Jobs can be created on a standby database using the `DBMS_SCHEDULER` package and can be associated with an appropriate database role such that they run when intended (e.g. when the database is the primary, standby, or both).

Switchover using SQL Apply with Oracle RAC databases no longer requires the prior shutdown of all but the first instance in each Oracle RAC cluster.

Data Guard SQL Apply parameters may also be set dynamically without requiring SQL Apply to be restarted. Using the `DBMS_LOGSTDBY.APPLY_SET` package, you can dynamically set initialization parameters, thus improving the manageability, uptime, and automation of a logical standby configuration.

Data Guard Broker Enhancements: The following enhancements further simplify management when using the Data Guard broker:

- Improved support for redo transport options, enabling an administrator to specify a connect description for Redo Transport Services.
- Elimination of database downtime when changing the protection mode to and from Maximum Availability and Maximum Performance.
- Support for single instance databases configured for HA using Oracle Clusterware as a cold failover cluster.

Enterprise Manager Grid Control 11g Enhancements: Enterprise Manager further simplifies management in the areas of:

- Creation of standby databases from existing RMAN backups
- Creation of an Oracle RAC standby database from an Oracle RAC primary.
- Automated standby clones for reporting, development, and test
- Automatic propagation of Enterprise Manager jobs and metric thresholds to the new primary database upon switchover or failover
- Fault-tolerant observer for Fast-Start Failover
- Enterprise Manager Data Recovery Advisor will utilize available standby databases when making recommendations for Intelligent Data Repair (IDR).

New Oracle Database 11*g* database options

Data Guard configurations can also be enhanced by utilizing new database options available for Oracle Database 11*g* Enterprise Edition. Database options are separately licensed products that extend the utility and performance of the Oracle Database. Two database options particularly relevant to Data Guard 11*g* are the Oracle Active Data Guard Option and the Oracle Advanced Compression Option.

NEW! **– The Active Data Guard Option enables read-only access to an up-to-date physical standby database. This can improve the performance of the production database, and makes a physical standby database an extension of the production environment, even while in standby role.**

Oracle Active Data Guard is an option for Oracle Database 11*g* Enterprise Edition that enhances Quality of Service by offloading resource-intensive activities from a production database to one or more synchronized standby databases. The Real-time Query feature included with the Active Data Guard Option enables read-only access to a physical standby database for queries, sorting, reporting, web-based access, etc., while continuously applying changes received from the production database.

Active Data Guard also enables RMAN block-change tracking on a physical standby, making it possible to perform fast incremental backups on a physical standby database. Tests have shown that incremental backups on a database with a moderate rate of change can complete up to 20x faster when using RMAN block-change tracking, compared to traditional incremental backups.

Oracle Advanced Compression is an option for Oracle Database 11*g* Enterprise Edition that helps manage growing amounts of data (that on average are tripling every two years) in a more cost effective manner. Advanced Compression compresses any type of data, including structured and unstructured data such as documents, images, and multimedia, as well as network traffic and data in the process of being backed up.

NEW! **– The Advanced Compression Option enables network compression when shipping log files needed to resolve gaps and resynchronize standby databases following network or standby server failures.**

The Advanced Compression Option will perform network compression of redo data during Data Guard 11*g* resolution of an archive log GAP on a standby database. This can accelerate the resynchronization of standby databases following network or standby database outages and more efficiently utilizes network bandwidth.

DATA GUARD PROCESS ARCHITECTURE

As shown in Figure 2, Data Guard uses several Oracle database processes to achieve the automation necessary for high availability and disaster recovery.

On the primary database, Data Guard uses a specialized background process, called LogWriter Network Server (LNS), to capture redo data being written by Log Writer and synchronously or asynchronously transmit the redo data to the standby database. The LNS process isolates Log Writer from the overhead of transmission and from network disruptions.

Synchronous redo transport (SYNC) requires the Log Writer on the primary database to wait for acknowledgement from LNS that the standby has received the

redo data and written it to a standby redo log before it can acknowledge the commit to the client application. This insures all committed transactions are on disk and protected at the standby location.

Asynchronous redo transport (ASYNC) does not require the Log Writer on the primary to wait for acknowledgment from the standby database that redo has been written to disk; commits are acknowledged to the client application asynchronous of redo transport.

NEW! **- Asynchronous redo transport performance enhancements further improve network utilization and improve recovery point objectives.**

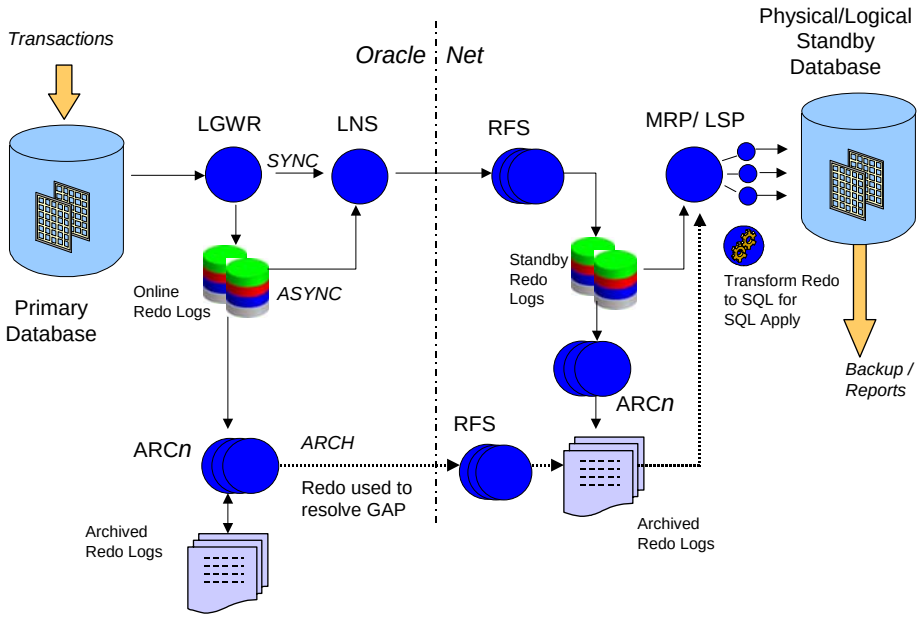

Figure 2: Data Guard Process Architecture

In addition, ASYNC very efficiently streams redo to the standby database to eliminate any overhead caused by network acknowledgment that would otherwise reduce network throughput.

In cases where the primary and standby databases become disconnected (network failures or standby server failures), and depending upon the protection mode chosen (protection modes are discussed later in this paper), the primary database will continue to process transactions and accumulate a backlog of redo data that cannot be shipped to the standby until a new network connection can be established (referred to as an archive log gap). While in this state, Data Guard continually monitors standby database status, detects when connection is re-established, and automatically resynchronizes the standby database with the primary to return the configuration to a protected state as fast as possible. Archiver (ARC*n*) processes are always utilized to ship log files that are needed to resolve gaps. There are several options available to accelerate gap resolution.

- Using the Advanced Compression Option, network compression can be performed on archive logs transmitted to resolve gaps in order to reduce transmission time and make maximum use of available bandwidth.
- A maximum of thirty ARC*n* processes can be enabled if there are a large number of archive logs that need to be transferred. Twenty-nine of these ARC*n* processes may ship to remote locations; one ARC*n* process is always dedicated to local archival.
- If there is a backlog with fewer archive logs, but the logs are very large in size, you can configure up to five ARC*n* processes to ship the contents of a single log file in parallel.

Finally, a Data Guard standby database will proactively request a new copy of an archived log if the standby apply process detects a missing log file, or any defect or corruption in an archive log that it has received. An ARC*n* process is used to fulfill such requests.

On the receiving side of Data Guard Redo Transport Services, Data Guard uses one or more Remote File Server (RFS) processes on the standby database to receive redo data and write it to a file called a Standby Redo Log (SRL). The Managed Recovery Process (MRP) is used to coordinate the application of redo data to a physical standby database, and the Logical Standby Process (LSP) is used to coordinate the application of SQL-translated redo data to a logical standby database.

If the Data Guard Broker is enabled, Data Guard also uses the Data Guard Broker Monitor (DMON) and other process to manage and monitor the primary and standby databases as a unified configuration.

KEY TECHNOLOGY COMPONENTS

Data Guard Configuration

The primary and standby databases in a Data Guard configuration can run on a single node or in an Oracle RAC environment. Standby databases connect to a primary database over TCP/IP using Oracle Net Services. There are no restrictions on where the databases are located provided that they can communicate with each other. However, for disaster recovery, it is recommended that the standby databases be hosted at sites that are geographically separated from the primary site.

NEW! **Data Guard 11g has several options for deploying different CPU architectures, O.S. binaries and Oracle database binaries, on primary and standby systems. For example, the primary database may be on Windows, and the standby database may be on Linux.**

See MetaLink Note 413484.1 for latest capabilities and restrictions

Data Guard 11*g* provides increased flexibility for Data Guard configurations where the primary and standby systems may have different CPU architectures, operating systems (e.g. Windows and Linux), operating system binaries (32-bit/64-bit), and Oracle database binaries (32-bit/64-bit) – subject to the restrictions defined in MetaLink Note 413484.1. Two constant requirements for primary and standby databases are that the endian format must be the same, and the release of the Oracle Database also be the same, except during rolling database upgrades using logical standby databases.

Protection Modes

Data Guard provides three modes of data protection to balance cost, availability, performance, and data protection. These modes define the rules that govern the behavior of the Data Guard configuration, and can be set easily using any of the available management interfaces, e.g. using the following simple SQL statement on the primary database:

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

To determine the appropriate data protection mode, enterprises need to weigh their business requirements for data protection against user demands for system response time. The following table outlines the suitability of each mode from a risk of data loss perspective.

Protection Mode	Risk of Data Loss In the Event of Primary Database Failure	Redo Transport
Maximum Protection	Zero data loss	SYNC
Maximum Availability	Zero data loss – assuming that prior to the failure there was no disruption in synchronous communication while the primary database was committing transactions	SYNC
Maximum Performance	Minimal data loss – as little as a few seconds depending upon network bandwidth	ASYNC

The following sections describe these protection modes in more detail.

Maximum Protection

Maximum Protection offers the highest level of data protection. Maximum Protection synchronously transmits redo records to the standby database(s) using SYNC redo transport. The Log Writer process on the primary database will not acknowledge the commit to the client application until Data Guard confirms that

the transaction data is safely on disk on at least one standby server. Because of the synchronous nature of redo transmission, Maximum Protection can impact primary database response time. Configuring a low latency network with sufficient bandwidth for peak transaction load can minimize this impact. Financial institutions and lottery systems are two examples where this highest level of data protection has been deployed.

It is strongly recommended that Maximum Protection be configured with at least two standby databases so that if one standby destination fails, the surviving destination can still acknowledge the primary, and production can continue uninterrupted. Users who are considering this protection mode must be willing to accept that the primary database will stall (and eventually crash) if at least one standby database is unable to return an acknowledgement that redo has been received and written to disk. This behavior is required by the rules of Maximum Protection to achieve zero data loss under circumstances where there are multiple failures (e.g. the network between the primary and standby fails, and then a second event causes the primary site to fail). If this is not acceptable, and you want zero data loss protection but the primary must remain available even in circumstances where the standby cannot acknowledge data is protected, use the Maximum Availability protection mode described below.

Maximum Availability

Maximum Availability provides the next highest level of data protection while maintaining availability of the primary database. As with Maximum Protection, redo data is synchronously transmitted to the standby database using SYNC redo transport services. Log Writer will not acknowledge that transactions have been committed by the primary database until Data Guard has confirmed that the transaction data is available on disk on at least one standby server. However, unlike Maximum Protection, primary database processing will continue if the last participating standby database becomes unavailable (e.g. network connectivity problems or standby failure). The standby database will temporarily fall behind compared to the primary database while it is in a disconnected state. When contact is re-established, Data Guard automatically resynchronizes the standby database with the primary.

Because of synchronous redo transmission, this protection mode can impact response time and throughput. Configuring a low latency network with sufficient bandwidth for peak transaction load can minimize this impact.

Maximum Availability is suitable for businesses that want the assurance of zero data loss protection, but do not want the production database to be impacted by network or standby server failures. These businesses will accept the possibility of data loss should a second failure subsequently affect the production database before the initial network or standby failure is resolved and the configuration resynchronized.

Maximum Performance

Maximum Performance is the default protection mode. It offers slightly less data protection with the potential for higher primary database performance than Maximum Availability mode. In this mode, as the primary database processes transactions, redo data is asynchronously shipped to the standby database using ASYNC redo transport. Log Writer on the primary database does not wait for standby acknowledgement before it acknowledges the commit to the client application. This eliminates the potential overhead of the network round-trip and standby disk i/o on primary database response time. If any standby destination becomes unavailable, processing continues on the primary database and there is little or no impact on performance.

During normal operation, data loss exposure is limited to that amount of data that is in-flight between the primary and standby – an amount that is a function of the network's capacity to handle the volume of redo data generated by the primary database. If adequate bandwidth is provided, total data loss exposure is very small or zero.

Maximum Performance should be used when availability and performance on the primary database are more important than the risk of losing a small amount of data. This mode is suitable for Data Guard deployments in cases where the latency inherent in a customer's network is high enough to limit the suitability of synchronous redo transmission.

Automatic Gap Resolution - Handling Communication Failures

Data Guard can smoothly handle network connectivity problems that temporarily disconnect the standby database from the primary database. The exact behavior is dictated by the rules of the chosen Data Guard protection mode.

When the last standby database becomes unavailable, both Maximum Availability and Maximum Performance allow the primary database to continue processing transactions. When connectivity to the standby is re-established, the accumulated archive logs are automatically shipped and applied to the standby database, quickly resynchronizing it with the primary. Note that the primary will also ship the current redo stream in parallel with the resynchronization process to prevent the standby database from falling any further behind the primary while the gap is resolved. The resynchronization process does not require any administrative intervention as long as the archive logs required to resynchronize the standby database are available on-disk at the primary database.

Apply Services - Redo Apply and SQL Apply

A standby database is initially created from a backup copy of the primary database. Once created, Data Guard automatically synchronizes the primary database and all standby databases within a Data Guard configuration by transmitting primary database redo data to the standby system(s) and then applying the redo data to the standby database.

Data Guard provides two methods to apply this redo data to the standby database, and these methods correspond to the two types of standby databases supported by Data Guard:

- Redo Apply, used for physical standby databases
- SQL Apply, used for logical standby databases

There is no distinction between these two types of standby databases as far as redo data transmission from the primary is concerned. Once this redo data reaches the standby server, it is the method used to apply redo data to the standby database that distinguishes these two types of standby databases.

Physical Standby Database – Redo Apply

NEW! **Redo Apply performance has been enhanced significantly for all workload profiles compared to Oracle Database 10g**

A physical standby database applies redo data received from the primary using Oracle media recovery. It is physically identical to the primary database on a block-for-block basis, and thus, the database schemas, including indexes, are the same.

How Redo Apply works

Data Guard Redo Apply uses a specialized process, called the Managed Recovery Process (MRP). As the RFS process is writing redo data to Standby Redo Logs (SRLs), MRP reads the redo data and applies it to the physical standby database.

MRP is started on the physical standby database by mounting the database and using the following command:

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

MRP may also transparently switch to reading from a standby archived log if the SRL is archived before MRP can complete reading of the SRL (a situation which may occur when the primary database has a very high redo generation rate).

MRP can be run in parallel for the best performance of Data Guard Redo Apply. MRP automatically determines the optimal number of parallel recovery processes at startup.

Data Validation Before Redo is Applied to the Standby Database

One of the significant advantages of Data Guard is its ability to use Oracle processes to validate redo data before it is applied to the standby database. Data Guard is a loosely coupled architecture where standby databases are kept synchronized by applying redo blocks, completely detached from possible data file corruptions that can occur at the primary database. In the case of SYNC redo transport, redo is shipped from primary SGA, and thus is completely detached from physical I/O corruptions on primary. The software code-path executed on standby database is also fundamentally different from that of primary – effectively secluding the standby database from software errors that can impact the primary database.

Corruption-detection checks occur at the following key interfaces:

- On the primary database during Redo Transport: `LGWR, LNS, ARCH`
- On the standby database during Redo Apply: `RFS, ARCH, MRP, DBWR`

If redo corruption is detected by Redo Apply at the standby database, Data Guard will re-fetch valid logs as part of archive log gap handling in the hope that the original archive log is free of corruption.

NEW! **Data Guard Redo Apply also detects lost writes caused by faulty storage hardware and firmware – eliminating yet another source of data corruption from impacting the standby database.**

Physical Standby also utilizes the new parameter:

`DB_LOST_WRITE_PROTECT.`

A lost write occurs when an I/O subsystem acknowledges the completion of a write, while in fact the write did not occur in the persistent storage. On a subsequent block read, the I/O subsystem returns the stale version of the data block, which can be used to update other blocks of the database, thereby corrupting it. When the `DB_LOST_WRITE_PROTECT` initialization parameter is set, the database will record buffer cache block reads in the redo log and this information can be used to detect lost writes.

Lost write detection is most effective when used with Data Guard. In this case, you set `DB_LOST_WRITE_PROTECT` in both primary and standby databases. When a standby database applies redo during managed recovery, it reads the corresponding blocks and compares the SCNs with the SCNs in the redo log, and:

- If the block SCN on the primary database is lower than on the standby database, it detects a lost write on the primary database and throws an external error (`ORA-752`). The recommended procedure to repair a lost write on a primary database is to failover to the physical standby and recreate the primary.
- If the SCN is higher, it detects a lost write on the standby database and throws an internal error (`ORA-600 3020`). To repair a lost write on a standby database, you must re-create the standby database or affected files.

In both cases, the standby database will write the reason for the failure in the alert log and trace file.

Taken together, the above capabilities deliver on a fundamental principle of Data Guard that corruptions that occur at the primary database should not impact the integrity of the standby database

NEW! **Real-time Query, a feature included with the Active Data Guard Option, enables read-only access to an up-to-date physical standby database to offload the overhead of queries and reporting from the primary database**

Redo Apply and Real-time Query

Using the Real-time Query feature of the Active Data Guard Option, a physical standby database can be opened read-only while redo apply is on (recovery is active) enabling queries to be run against an up-to-date replica of the primary database. This is enabled using Enterprise Manager Grid Control 11*g*, or manually from the command line in the following manner:

- To open a standby database for read-only access when redo apply is active, first cancel redo apply using the following statement:
  ```
  SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
  ```
- Then, open the database for read-only access using the following statement:
  ```
  SQL> ALTER DATABASE OPEN;
  ```
- Once the standby database has been opened, restart redo apply using the following command.
  ```
  SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE
  ```

NEW! **A snapshot standby database can be open read-write and used for test and other purposes. The primary database continues to ship redo data to the snapshot standby to insure that data remains protected at the remote location. The snapshot standby will archive this redo for safekeeping. When test activities are complete, the snapshot standby is resynchronized with the primary by converting it to a standby database and applying the log files that it had received from the primary while it was operating as a Snapshot Standby.**

Snapshot Standby

A snapshot standby database is a fully updateable standby database that is created by converting a physical standby database into a snapshot standby database. A snapshot standby database receives and archives, but does not apply, redo data from a primary database. Redo data received from the primary database is applied when a snapshot standby database is converted back into a physical standby database, after discarding all local updates to the snapshot standby database.

A snapshot standby can be created from Enterprise Manager, the Data Guard Broker command line interface (`DGMGRL`) or from SQL*Plus. To create a snapshot standby from SQL*Plus, simply issue the following command on the physical standby database:

```
SQL> ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;
```

When the standby database is converted to a snapshot standby, an implicit guaranteed restore point is created and flashback database is enabled. After you are done using the snapshot standby, it can be converted back to a physical standby database and resynchronized with the primary by issuing:

```
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY DATABASE;
```

Data Guard implicitly flashes the database back to the guaranteed restore point and automatically applies the primary database redo that has been archived by the standby since the snapshot was created. The guaranteed restore point is dropped once this process completes.

Logical Standby Database – SQL Apply

A logical standby database contains the same logical information as the primary database, although the physical organization and structure of the data can be different. The SQL Apply technology keeps the logical standby database synchronized with the primary database by transforming the redo data received from the primary database into SQL statements and then executing the SQL statements on the standby database. Similar to a physical standby, this makes it possible for the logical standby database to be accessed for queries and reporting purposes at the same time apply is active.

NEW! **SQL Apply support added for:**
- XMLType data type (CLOB)
- Parallel execution of DDL
- Transparent Data Encryption (TDE)
- DBMS_FGA (Fine Grained Auditing)
- DBMS_RLS (Virtual Private Database)
- DBMS_SCHEDULER (Scheduler)

A logical standby database has greater flexibility than a physical standby, deriving from the fact that it is updated using SQL statements. This enables a logical standby database to be open in read-write mode and able to perform other tasks that require read-write access to the database (such as adding local tables that only exist in the standby database, or performing reporting or summations that require read-write access). These tasks can be optimized by creating additional indexes and materialized views on the tables maintained by SQL Apply (note that though the database is open read-write, SQL Apply will not allow changes to data that it is responsible for synchronizing with the primary database). A logical standby database can host multiple database schemas, and users can perform normal data manipulation operations on tables in schemas that are not updated from the primary database.

A logical standby database has some restrictions on datatypes, types of tables, and types of DDL and DML operations. Please refer to the documentation [2] for a list of these unsupported datatypes and storage attributes.

How SQL Apply works

NEW! **SQL Apply performance has been enhanced for workloads characterized by inserts and updates to tables that are not partitioned and do not contain LOB, LONG or XML type column. Parallel DDLs are also applied in parallel by SQL Apply further enhancing apply performance.**

SQL Apply uses a collection of background processes that perform the task of applying changes from the primary database to the logical standby database. Figure 3 shows the flow of information and the role that each process performs.

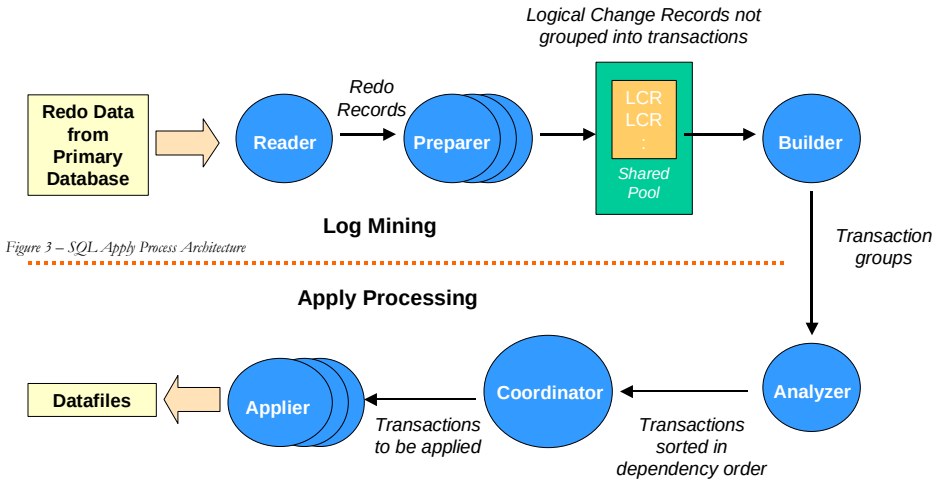

Figure 3 – SQL Apply Process

Following is a summary of the functionalities of each of the processes:

- The *Reader* process reads incoming redo records from standby redo logs.

- The *Preparer* processes convert the block changes into table changes, or logical change records (LCRs).
- The *Builder* process assembles completed transactions from individual LCRs.
- The *Analyzer* process examines the completed transactions, identifying dependencies between the different transactions.
- The *Coordinator* process (also known as the Logical Standby Process, or LSP), assigns transactions to the apply processes, monitors dependencies between transactions, and authorizes the commit of changes to the logical standby database.
- The *Applier* processes apply the LCRs for the assigned transaction to the database and commit the transactions when instructed by the Coordinator.

Entering this SQL command starts these SQL Apply processes:

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

Similar to Redo Apply, corruption detection checks occur at key interfaces:

- On the primary database during Redo Transport: `LGWR, LNS, ARCH`
- On the standby database during SQL Apply: `RFS, ARCH, SQL, LSP, DBWR`

How to decide which to use – Redo or SQL Apply

Both approaches have much in common. Physical and logical standby databases utilize the same redo transport and role management services, only the apply process is different. Either can be used to offload query and reporting from the primary database. Either can be used to execute a rolling database upgrade (physical standby users utilize a transient logical standby as described above). All standby databases are first created as physical standby databases during the standby database instantiation process. Several additional steps are required to convert a physical standby database to a logical standby database.

In practice, users tend to choose between physical or logical standby based on the following reasons:

Considerations that influence the decision to utilize Redo Apply

- Using the Active Data Guard Option, a physical standby database can be easily utilized to offload the production database of the overhead of processing read-only queries and reports.
- Very high performance requirements – the redo apply process used by physical standby is very efficient thus consumes less I/O and CPU compared to SQL Apply. The efficiency of the redo apply process makes it practical to host multiple production and/or standby databases on the same server , something that resource consumption may make impractical when using SQL Apply. This can be important for companies that wish to reduce cost by deploying shared resources to host multiple standby databases.

- Relative simplicity. Redo Apply is a simpler process, appealing to users who may prefer a simpler solution.
- A physical standby can become a Snapshot Standby for test, development, or other purposes.
- A physical standby can be used to offload the primary host from doing backups since it is an exact replica of the primary database.
- A physical standby has no restrictions. Its operation and performance is completely transparent to data types or transaction profile – redo is redo.
- In a disaster recovery scenario, some users prefer their standby database to be an exact physical replica of the primary.

Considerations that influence the decision to utilize SQL Apply:

- A logical standby database has greater flexibility deriving from the fact that is open read-write. For example, some reporting applications require read-write access to the database, and although physical standby users can work around this restriction by using database links to another database, logical standby enables read-write access without any additional steps. Because the logical standby tables that are protected by Data Guard can be stored in a different physical layout than on the primary database, additional indexes and materialized views can also be created to improve query performance.
- A logical standby database is more easily used for staging and processing of data for data warehouse applications, decision support, or to populate data marts. A logical standby can host additional database schemas beyond the ones that are protected by the Data Guard configuration, and users can perform DDL or DML operations on those schemas any time.

Managing a Data Guard Configuration

System Views. Data Guard offers various views to monitor the run-time performance of the Data Guard configuration in a granular fashion. These can be accessed through SQL*Plus, Data Guard Broker, or Enterprise Manager Grid Control.

NEW! **Response time histogram to guide users on the optimum value of `NET_TIMEOUT` based upon data from their production environment**

For example, `V$REDO_DEST_RESP_HISTOGRAM,` is a fixed view containing a histogram of response times for SYNC redo transport destinations. The data in this view is very useful when determining the appropriate value for the network timeout attribute used for synchronous destinations. Used for Maximum Availability mode, the `LOG_ARCHIVE_DEST_n NET_TIMEOUT` attribute specifies the maximum period that Log Writer will wait for an LNS acknowledgement before giving up on the remote destination and allowing Log Writer to continue. Setting too low a value can result in frequent timeouts, reducing data protection levels. Too high a value can negatively impact primary database throughput by stalling the database when it should give up on the remote destination. Data from this histogram gives

administrators the information needed to optimize the tradeoff between HA and data protection objectives.

Data Guard Broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations. All management operations can be performed either through Oracle Enterprise Manager Grid Control, which uses the Broker, or through the Broker's specialized command-line interface (`DGMGRL`).

The following list describes some of the operations that the Broker automates and simplifies:

- Creating and enabling Data Guard configurations
- Managing an entire Data Guard configuration from any site in the configuration.
- Invoking switchover or failover operations (including Fast-Start Failover) that involve complex role changes across all systems in the configuration.
- Monitoring apply rates, capturing diagnostic information, and detecting problems quickly with centralized monitoring, and event notification.

Enterprise Manager Data Guard management pages and wizards further simplify creating and managing a Data Guard configuration. Enterprise Manager enables historical trend analysis on the Data Guard metrics that it monitors – for example, how the metric's performance has been in the last 24 hrs, or last 5 days, etc. Also, through Enterprise Manager, it is possible to set up notification-alarms such that administrators may be notified in case the metric crosses the configured threshold value. The screenshot in Figure 4 shows the Data Guard home page in Enterprise Manager.

Examples of Data Guard metrics monitored by Enterprise Manager are:

- Estimated Failover Time – The approximate number of seconds it would require to failover to this standby database.
- Apply Lag – Shows how far the standby is behind the primary.
- Redo Apply Rate – The rate at which redo is applied on the standby.
- Redo Generation Rate – The rate at which redo is generated on the primary.
- Transport Lag – The approximate number of seconds of redo not yet available on this standby database. This may be because the redo has not yet been shipped or there may be a gap.
- Data Guard Status – Shows the status of each database in the Data Guard configuration.

- Fast-Start Failover Occurred – When fast-start failover is enabled, this metric will generate a critical alert on the new primary database (old standby) if a fast-start failover occurs.
- Fast-Start Failover Time – When fast-start failover is enabled, this metric will generate a critical alert on the new primary database (old standby) if fast-start failover occurs, indicating the time stamp of the occurrence.

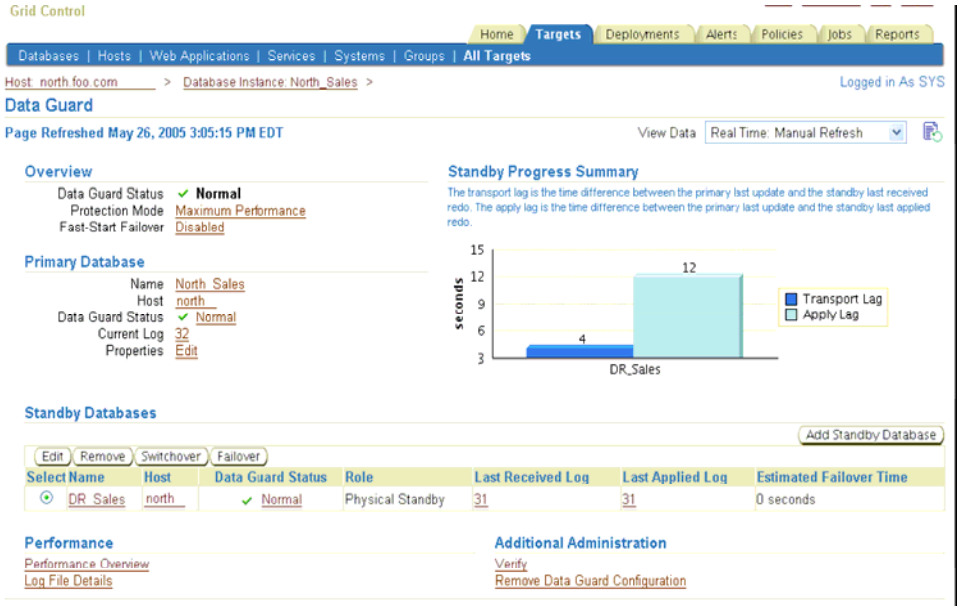

Figure 4 - Data Guard Home page in Oracle Enterprise Manager

Role Management Services

Data Guard Role Management Services quickly transition a designated standby database to the production role for planned (switchover) and unplanned (failover) outages of the production site. Role Management Services also make it very straightforward to test disaster recovery preparedness.

Switchover

A switchover is used to reduce primary database downtime during planned outages, such as operating system or hardware upgrades, or rolling upgrades of the Oracle database software and patch sets.

A switchover operation requires all user sessions to be disconnected from the primary database. Following that, the primary database is transitioned to the standby role, after which the standby database is transitioned to the primary role.

A switchover is initiated by the administrator using the Oracle Enterprise Manager GUI interface, the Data Guard Broker's command line interface, or directly through SQL.

For example – the following single Data Guard Broker CLI (`DGMGRL`) command initiates and completes the switchover to the standby database "Chicago":

```
DGMGRL> SWITCHOVER TO Chicago;
```

Once initiated, Data Guard automates the actual role transition processes. No data is lost in the process.

Failover

Failover is the operation of bringing one of the standby databases online as the new primary database when an unplanned failure occurs on the primary database. This enables recovery time objectives to be achieved by quickly promoting the standby to the primary role instead incurring downtime while the events that impact the primary are diagnosed and resolved. A failover operation does not require the standby database to be restarted. Also, as long as the database files on the original primary database are intact, and the database can be restarted, the original primary can be reinstated and resynchronized as a standby database for the new primary using Flashback Database – it does not have to be restored from a backup.

Manual failover is initiated by the administrator using the Oracle Enterprise Manager GUI interface, the Data Guard Broker's command line interface, or directly through SQL*Plus. on the standby database that will assume the primary role. For example – the following single Data Guard Broker CLI (`DGMGRL`) command initiates and completes the failover to the standby database "Chicago":

```
DGMGRL> FAILOVER TO Chicago;
```

A manual failover operation ensures zero data loss if Data Guard was being run in the Maximum Protection or Maximum Availability and the target standby database was synchronized at the time of the failover. In Maximum Performance mode, if there were still some redo data in the primary database that had not yet been sent to the standby at the time of failover – that data may be lost. In addition to manual failover, customers have the option to configure Data Guard to perform automatic failover in a very controlled manner, discussed further in the section below.

Fast-Start Failover

NEW! **Fast-Start Failover (automatic failover) also supports Data Guard configurations in Maximum Performance (ASYNC Redo Transport) through use of a user-configurable data loss threshold**

Fast-Start Failover allows Data Guard to automatically fail over to a previously chosen, standby database without requiring any manual steps to invoke the failover. Further, upon return of the failed primary, it is automatically reinstated into the configuration as a standby of the new primary database. Fast-Start Failover can be used only in a Data Guard Broker configuration and can be configured only through DGMGRL or Enterprise Manager.

The fast-start failover configuration is monitored by a separate Observer process, which is a lightweight process integrated in the DGMGRL client-side component. It continuously monitors the fast-start failover environment to ensure the primary database is available. If both the Observer and the standby database lose connectivity to the primary database, the Observer attempts to reconnect to the primary database for a configurable amount of time before initiating a fast-start failover. Fast-start failover is designed to ensure that out of the three fast-start failover members – the primary, the standby and the Observer, at least two members agree to major state transitions, thus avoiding conditions such as split-brain scenarios in which there may be two divergent primary databases serving production workload. The simple, yet elegant architecture of fast-start failover makes it an excellent candidate to be used in high availability situations where data protection is also important.

A benefit of utilizing Enterprise Manager is that it also enables high availability for the Observer in a Data Guard Fast-Start Failover configuration. Enterprise Manager can monitor the status of the Observer, detect Observer failure, attempt to restart the Observer on its original host, and if that fails will restart the Observer on a previously designated second host. Enterprise Manager's ability to insuring high availability for the Observer is a critical element of any HA strategy that utilizes Data Guard Fast-Start Failover.

Flexible Configuration Options

The administrator can restrict automatic failover to cases where there will be no data loss using Maximum Availability, or cases where the amount of data loss will not exceed a previously configured threshold using Maximum Performance.

The following are among the properties provided to control the behavior of fast-start failover:

- `FastStartFailoverThreshold` - specifies the number of seconds the observer and target standby database will wait (after detecting the primary database is unavailable) before initiating a failover.
- `FastStartFailoverLagLimit` - is a property used to specify a time-based limit to the amount of data loss allowed in an automatic failover. If the standby database's applied redo point is within this many seconds of the primary's redo generation point, a fast-start failover will be allowed. If its applied point lags beyond this limit, a fast-start failover is not allowed. This property is only applicable to Maximum Performance; it is not used if fast-start failover is enabled when the configuration is operating in Maximum Availability mode.
- `FastStartFailoverAutoReinstate –` this property can be set to FALSE to prevent automatic reinstatement of a failed primary after a failover has occurred if you wish to perform diagnostic and repair on the old primary.

NEW! **Fast-start Failover provides user configurable events that when detected by the observer, will trigger immediate failover without waiting for the fast-start failover threshold to expire**

Oracle Enterprise Manager or `DGMGRL ENABLE FAST_START FAILOVER CONDITION` commands can also be used to specify conditions for which a fast-start failover should occur without waiting for the failover threshold time period to expire. They include:

- "Datafile Offline" Data file offline due to a write error.
- "Corrupted Controlfile" Corrupted controlfile.
- "Corrupted Dictionary" Dictionary corruption of a critical database object.
- "Inaccessible Logfile" Log Writer is unable to write to any member of a log group due to I/O error.
- "Stuck Archiver" Archiver is unable to archive a redo log because device is full or unavailable.

Finally, the `DBMS_DG PL/SQL` package can be used to allow an application to request a fast-start failover when it encounters specific conditions. When a condition uniquely known to the application is detected, it may call the `DBMS_DG.INITIATE_FS_FAILOVER` procedure, thus alerting the primary database that it wants a fast-start failover to occur immediately. The primary database will notify the observer of this and the observer will immediately initiate a fast-start failover, assuming the standby is in a valid fast-start failover state (observed and either synchronized or within lag) to accept a failover.

RESTORING OLD PRIMARY AS A NEW STANDBY

Following a failover, the administrator will need to reinstate the old primary as a new standby in the Data Guard configuration, to resume data protection capabilities of the configuration.

If the old primary can be restarted and the database can be mounted with its data files intact, then Flashback Database can be used to reinstate the old primary as a standby database to the new primary, without needing to recreate it from a backup of the new primary (Flashback Database must be enabled before the failure event occurs). As described above, this reinstatement is automatic when using Fast-Start Failover.

In the case of a manual failover, this is accomplished simply by executing a flashback the old primary database to the point in time when the failover occurred (given by the `STANDBY_BECAME_PRIMARY_SCN` column in the `V$DATABASE` view), starting the database as a standby database and then letting Data Guard automatically synchronize the new standby database with the new primary database. To easily accomplish this using `DGMGRL`, simply restart the old primary database and then issue the following command:

```
DGMGRL> REINSTATE DATABASE 'database';
```

AUTOMATIC CLIENT FAILOVER

Failovers can be sorted into one of the following broad categories:

- Complete-site failover utilizes a secondary site to host a Data Guard standby database and a completely redundant set of middle-tier application servers. When failing over to the secondary site, the middle-tier servers are started and a network load balancer is redirected to the new primary site. A Data Guard failover will transition the standby database at the secondary location to the primary role. The Oracle MAA paper, *Oracle Database High Availability Best Practices* [5], provides guidance for implementing complete site failover.
- When a node within an Oracle RAC fails, clients attached to the failed node must be quickly notified that a failure has occurred, and must reconnect to the surviving nodes in the cluster to continue processing. The technical white paper, *Workload Management with Oracle Real Application Clusters* [6], provides guidance for handling node failures within an Oracle RAC.
- Partial-site failover occurs when the primary database has become unavailable but the primary site remains intact, and affected clients must be redirected to a new primary database at the secondary location following a Data Guard failover. An overview of implementing client failover for partial-site failure is provided below. For complete details, please refer to *Client Failover Best Practices for Highly Available Oracle Databases* [7].

At a high level, automating client failover in a Data Guard configuration includes relocating *Database Services* to the new primary database as part of a Data Guard failover,, notifying clients that a failure has occurred to break them out of TCP timeout, and redirecting clients to the new primary database.

Service Relocation

Oracle Database 10*g* introduced an automatic workload management facility for Oracle RAC and single instance (non-RAC) databases, called Database Services [8], that enable you to group database workloads and easily designate computing resources to service that workload. You can define a database service for a particular application, such as the application `'sales'` used in the example below, and should the primary database offering the service fail, the service can be automatically relocated to the new primary database a part of a Data Guard failover.

Within an Oracle RAC cluster users are able to access a service independent of the instance providing it because, using listener registration, all listeners in a cluster are aware of which instance is currently providing the database service when a connection request is received. If the instance providing the service fails, Oracle RAC quickly relocates the service to a surviving instance within the cluster.

The same concept of service relocation applies to a Data Guard configuration. To illustrate how this occurs, assume a simple Data Guard configuration with single node primary and standby databases, and a database service named `'sales'` that

has been created and configured with all of the appropriate high availability settings [7]. Client applications connect to the service `'sales'` using an Oracle Net alias that includes all hosts, both primary and standby, in the configuration. Unwanted connection attempts to the standby database are prevented because the service name used in the Oracle Net alias only runs on the instances for the primary database. Relocation of the service to the new primary is automated by use of a trigger that fires when a Data Guard role change transitions the standby database to the primary role, starting the service on the new primary database. For example:

```
CREATE OR REPLACE TRIGGER manage_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('sales');
   ELSE
     DBMS_SERVICE.STOP_SERVICE('sales');
   END IF;
END;
```

Client Notification and Reconnection

Continuing with the example above, client notification and reconnection prevents clients that are connected to the original primary at the time of failure from falling into a hung-state waiting for lengthy TCP timeouts to expire. Oracle will notify these clients that a failure has occurred, break them out of TCP timeout, and have both new and existing connections directed to surviving RAC nodes, or to the new primary database following a Data Guard failover.

This is accomplished for OCI clients using Fast Application Notification (FAN). The new primary knows which clients were connected to the failed instance and as part of the failover process will notify them to reconnect to the service `'sales'`, used in the above example.

JDBC clients in a Data Guard configuration are notified using Fast Connection Failover and a trigger created on the standby database that fires on the `DB_ROLE_CHANGE` system event [7] when the standby database transitions to the primary role. The trigger calls a publisher program provided by Oracle [7] that notifies JDBC clients that the database service is available on the new primary, breaking stalled clients out of their TCP timeout.

The final task is to insure that once clients have been notified of the failure, that they are not subject to TCP timeouts if they subsequently attempt to reconnect to a failed host. This is accomplished using SQLNET outbound connect timeouts such that clients will quickly attempt to connect to the next host in an address list if the first is not available [7].

Additional configuration details for OCI, OLE DB, and JDBC clients along with examples of triggers used to relocate database services and publish HA events for

JDBC clients are described in the MAA paper, *Client Failover Best Practices for Highly Available Oracle Databases* [7]

Role Transition Events

The Data Guard `DB_ROLE_CHANGE` system event is fired whenever a database transitions from one role to another. This is much like the `ON STARTUP` system event, except that it fires after a role change. Administrators can develop triggers that execute when this event occurs as a way to manage post role-change tasks. The event fires when the database opens for the first time after the role transition regardless of its new role.

The `DB_ROLE_CHANGE` system event can be used to manage/automate post role change tasks and is one method for facilitating automatic client failover. Typical tasks could include starting a service / services on the new production database, changing name resolution services or connection descriptors so clients will reconnect to the new production database, starting third party applications, adding temporary tablespaces, and so on. `DB_ROLE_CHANGE` is a flexible mechanism to allow the administrator to automate any actions that can be accomplished via database trigger.

ROLLING DATABASE UPGRADES

NEW! **Physical Standby users can execute SQL Apply rolling upgrades without requiring additional disk for a logical standby database. Simply converting the physical to a logical, execute the rolling upgrade, and then revert the logical back to a physical standby database – using the `KEEP IDENTITY` clause**

Oracle Database software upgrades for major release and patchset upgrades (10.1.0.3 onwards) can be performed in a rolling fashion – with near zero database downtime, by using Data Guard SQL Apply (see Figure 5).

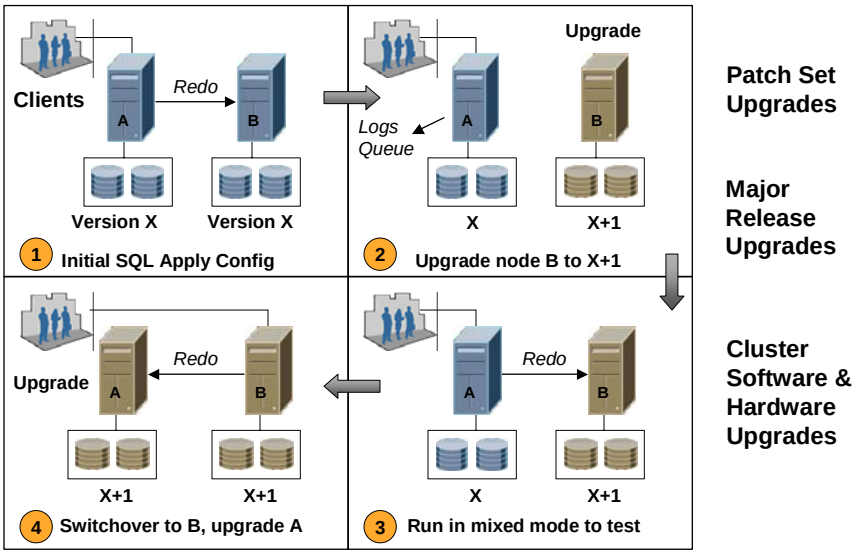

Figure 5 – Rolling Database Upgrades with SQL Apply

The steps in a rolling upgrade process involve upgrading the logical standby database to the next release, running in a mixed mode to test and validate the upgrade, doing a role reversal by switching over to the upgraded database, and then finally upgrading the old primary database. While running in a mixed mode for testing purposes, the upgrade can be aborted and the software downgraded, without data loss. For additional data protection during these steps, a second standby database may be used.

Beginning with Oracle Database 11*g*, a physical standby database can also take advantage of the rolling upgrade feature provided by a logical standby. Through the use of the new `KEEP IDENTITY` clause option to the `SQL> ALTER DATABASE RECOVER TO LOGICAL STANDBY` statement, a physical standby database can be temporarily converted into a logical standby database for the rolling upgrade, and then reverted back to the original configuration of a primary database and a physical standby database when the upgrade is done.

CASCADED DESTINATIONS

Data Guard provides many flexible configuration options. Using Cascaded Destinations, a physical standby database can forward the redo it receives from the primary database to another standby database. Since the primary database sends redo data to the first standby databases, this feature reduces the load on the primary system, and can also reduce network traffic and use of valuable network resources at the primary site when multiple standby databases are required. Note that Oracle RAC and the Data Guard Broker are not supported in a Data Guard configuration that includes cascaded destinations.

DATA GUARD AND ORACLE REAL APPLICATION CLUSTERS

"Because of our familiarity with Oracle, Data Guard was a natural fit. We had the Oracle expertise. We did not have to buy a new grade of storage or a new component for our existing storage systems or train ourselves on a third party replication solution. Data Guard enables us to achieve a higher level of data protection and availability."

- Mike Balint
Senior Database Administrator
Burlington Coat Factory

Data Guard and Oracle RAC are complementary technologies providing the highest possible level of scalability, availability, and data protection. Except for a restriction that applies to the use of cascaded destinations (described above), the integration between Oracle RAC and Data Guard is seamless. Any combination of Oracle RAC and single node databases can participate and assume any role in a Data Guard configuration. Oracle RAC provides the ideal HA solution to protect against server failure simultaneous with providing industry unique capabilities for workload management and scalability. Data Guard provides an additional level data availability and protection with complete redundancy that minimizes downtime due to complete storage array failure, operator errors, certain planned maintenance that can not be done in rolling fashion across Oracle RAC nodes, or multiple and correlated failures that can result in site failure.

MAXIMUM AVAILABILITY ARCHITECTURE

Oracle Maximum Availability Architecture (MAA) [9] is an Oracle tested and customer validated best-practices blueprint for deploying Oracle high availability technologies. The goal of MAA is to remove the complexity from designing optimal high availability architectures.

MAA best practices include recommendations on various aspects of a Data Guard configuration, such as a configuration with Oracle RAC, optimizing redo transport, switchover/failover operations, client failover, Redo Apply performance, SQL Apply configuration and tuning, etc. Customers involved with Data Guard implementations are strongly recommended to refer to MAA best practice guidelines.

DATA GUARD AND REMOTE-MIRRORING SOLUTIONS

"We utilize EMC Symmetrix and we've got bandwidth, so we've got the ability to use solutions such as SRDF, but for this critical database system, we went with Data Guard. Data consistency and data integrity were the main the drivers."

– David Willen
Chief Technology Officer
BarnesandNoble.com

Remote-mirroring solutions are often seen as a way to offer simple and complete data protection. There are two kinds of remote mirroring solutions – (a) host-based replication, and (b) storage array-based mirroring.

In Host-based Replication solutions, specialized file system drivers or volume manager components in the primary server intercept local writes, package them in logical messages, and synchronously or asynchronously send them over IP to remote (or secondary) hosts. Such solutions need to maintain specialized logs to keep track of write-ordering. The data volumes on the secondary server cannot be used (even for read-only access) while replication is in progress.

In Storage Array-based Mirroring solutions, storage array controllers at the primary site mirror changed disk I/O blocks to a similar storage array at the secondary site. These changes are sent using protocols such as ESCON, FICON and Fibre Channel, although in some recent versions iSCSI and IP-based transport are also supported. The mirroring over the appropriate communication links is controlled by specialized link adapters loaded with appropriate firmware. As I/Os occur at the primary server, data is written to the cache of the source array, and placed in a queue. The link adapter takes the first entry of the queue and moves it across the link to the mirrored array.

Data Guard is inherently more efficient, less expensive, and better optimized for protecting Oracle databases than remote mirroring solutions. Customers do not need to buy or integrate a remote mirroring solution with Data Guard to protect an Oracle database. A significant added benefit of Data Guard is the ability to utilize the standby database for productive purposes while it is in standby role. For a detailed analysis on this topic, refer to [10].

DATA GUARD CUSTOMERS

Data Guard functionality was first available with Oracle Version 7 and has continued to add new functionality and become a more mature technology with each subsequent Oracle release. It is deployed for mission-critical applications at customer sites worldwide. A number of detailed implementation case studies are available on OTN [11].

CONCLUSION

"Our Trans-Atlantic deployment of Data Guard across 3,000 miles delivers a level of reliability and data protection required by the many regulatory agencies we are accountable to as a specialty pharmaceutical and biologics company operating in 27 different countries across several continents"

- Kevin Bradley, Associate Director Global R&D Business Systems, Support and Delivery

Data Guard 11*g* fundamentally changes the traditional disaster recovery paradigm by offering an integrated HA/DR solution with unparalleled data protection and where standby systems simultaneously support production and QA functions *while they are in standby role*.

Oracle Data Guard is a comprehensive data protection, disaster recovery and high availability solution for the enterprise. It offers a flexible and easy-to-manage framework that addresses both planned and unplanned outages. Physical and logical standby databases provide high-value data protection while offloading overhead from primary databases. The various data protection modes provide flexibility to adapt to different levels of protection, performance and infrastructure requirements. The Data Guard Broker in combination with Oracle Enterprise Manager provides an easy-to-use configuration and management framework.

Regardless of the length to which high-availability has previously been built into an IT infrastructure using clusters, disk mirroring, and various backup and recovery strategies, it is a fact that data protection, availability, and your return on your IT investment is universally enhanced by the addition of Data Guard to your IT architecture.

REFERENCES

1. Oracle Data Guard
   http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html

2. Oracle Data Guard Concepts and Administration
   http://download.oracle.com/docs/cd/B28359_01/server.111/b28294/toc.htm

   Oracle Data Guard Broker
   http://download.oracle.com/docs/cd/B28359_01/server.111/b28295/toc.htm

3. Overview: Oracle Database High Availability
   http://www.oracle.com/technology/deploy/availability

4. Using RMAN with Data Guard – MAA Best Practices
   http://www.oracle.com/technology/deploy/availability/pdf/RMAN_DataGuard_10g_wp.pdf

5. Oracle High Availability Architecture and Best Practices
   http://download-west.oracle.com/docs/cd/B19306_01/server.102/b25159/toc.htm

6. Workload Management with Oracle Real Application Clusters (Provides a detailed explanation of the implementation of Services, FAN and Fast Connection Failover)
   http://www.oracle.com/technology/products/database/clustering/pdf/twpracwkldmgmt.pdf

7. Client Failover Best Practices for Highly Available Oracle Databases
   http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf

8. Database Services
   http://download-west.oracle.com/docs/cd/B19306_01/rac.102/b14197/hafeats.htm

9. Oracle Maximum Availability Architecture,
   http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

10. The Right Choice for Disaster Recovery: Data Guard, Stretch Clusters or Remote Mirroring,
    http://www.oracle.com/technology/deploy/availability/techlisting.html

11. Oracle High Availability Case Studies,
    http://www.oracle.com/technology/deploy/availability/htdocs/HA_CaseStudies.html

Oracle Data Guard 11*g*, The Next Era in Data Protection and Availability
August 2007
Authors: Ashish Ray, Larry Carpenter, Joseph Meeks
Technical Review: Serge De La Sablonniere, Mike Dietrich, Holger Kalinowski, Michael Rhys

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com