

SQL Apply Best Practices:  
Oracle Data Guard 10g Release 2  
*Oracle Maximum Availability Architecture White Paper*  
*April 2007*

# Maximum Availability Architecture

Oracle Best Practices For High Availability

## SQL Apply Best Practices: Oracle Data Guard 10g Release 2

Executive Overview .....	4
Benefits of SQL Apply.....	4
Efficient Use of Standby Hardware Resources.....	4
Reduction in Primary Database Workload .....	4
Introduction .....	5
SQL Apply Enhancements .....	5
Oracle Database 10g Release 1 Enhancements .....	5
Standby Redo Log Files.....	5
Real-Time Apply.....	5
Support for Maximum Protection Mode.....	5
Zero Downtime Instantiation .....	6
Improved SQL Apply Performance.....	6
Faster SQL Apply Switchover and Failover.....	6
Identification of Unnecessary Log Files .....	6
Integration with Flashback Database to Resolve Logical Failures....	6
Support for More Data Types .....	7
Ability to Upgrade the Oracle Database Using SQL Apply .....	7
Improved Redo Data Transmission Security .....	7
Oracle Database 10g Release 2 Enhancements .....	7
Improved SQL Apply Performance.....	7
Support for Fast-Start Failover .....	7
Enhanced Data Guard Statistics .....	7
Log Writer Process Asynchronous Transport Enhancements.....	7
Modified SQL Apply Parameters for Improved Manageability .....	8
Simpler Logical Standby Configuration with Less Tuning .....	8
Support for More Data Types .....	8
SQL Apply Best Practices .....	8
Confirm Data Type Support.....	9
Use Standby Redo Log Files .....	10
Use Real-Time Apply .....	10
Consider Using a No Data Loss Protection Mode.....	10
Tune SQL Apply.....	11
Adjust How Transactions Are Applied on the Logical Standby Database .....	11
Use the Standard Oracle Tuning Methodology .....	13

- Additional Tuning Guidelines Specific to Logical Standby
  - Databases..... 14
  - Benefit From Improved SQL Apply Performance..... 14
    - Performance Improvement: Increasing MAX\_SERVERS ..... 16
    - Performance Improvement: Changing COMMIT\_ORDER ..... 16
  - Implement Faster SQL Apply Switchover and Failover..... 17
  - Flash Recovery Area and Standby Archived Log Files ..... 17
  - Manage Logical Standby Archived Log Files ..... 18
  - Use Fast-Start Failover..... 18
  - Use Enhanced Data Guard Management Statistics ..... 19
  - Use the Asynchronous Redo Transport Enhancements..... 19
- Conclusion..... 20
- Appendix A ..... 21
  - Quick Reference - Upgrading from Oracle9i Database to Oracle Database 10g..... 21
- References ..... 21

## SQL Apply Best Practices: Oracle Data Guard 10g Release 2

### EXECUTIVE OVERVIEW

With [Oracle Data Guard](#)[1] SQL Apply in Oracle Database 10g, Oracle is addressing the requirements of the business community for an online disaster-recovery solution that also provides a means to offload reporting and decision support operations from the primary database.

A logical standby database contains the same logical information as the production database, although the physical organization and structure of the data can be different. A logical standby database is kept synchronized with its primary database through SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database.

### Benefits of SQL Apply

Data Guard SQL Apply and logical standby databases provide the following benefits:

#### Efficient Use of Standby Hardware Resources

In addition to disaster-recovery requirements, you can use a logical standby database for other business purposes. It can host additional database schemas beyond the ones that are protected in a Data Guard configuration, and users can perform normal data definition language (DDL) or data manipulation language (DML) operations on those schemas any time. Because SQL Apply maintains tables in a different physical layout from how they are stored on the primary database, you can create additional indexes and materialized views on the logical standby database to improve query performance and satisfy specific business requirements.

#### Reduction in Primary Database Workload

A logical standby database can remain open while SQL Apply updates its tables, and those tables are simultaneously available for read access. Therefore, a logical standby database is an excellent choice for performing queries, summations, and reporting activities, thereby offloading the primary database from those tasks and saving valuable CPU and I/O cycles on the primary database host.

## INTRODUCTION

This white paper describes the characteristics of SQL Apply and logical standby databases, and describes the best practices for configuring and managing them when using Oracle Database 10g Release 2.

Updates to this white paper can be found on the [Maximum Availability Architecture \(MAA\)](#)[2] Web page.

In addition, you can reference this paper's complementary MetaLink Note [312434.1](#)[3], "Oracle 10g Data Guard SQL Apply Troubleshooting." There is also MetaLink Note [274170.1](#)[4], "Logical Standby Master Index Page," with links to other support notes that are directly relevant to SQL Apply. Together, these documents provide practical advice on best practices for configuring, managing, tuning, and troubleshooting SQL Apply and logical standby databases.

## SQL APPLY ENHANCEMENTS

This section lists the Data Guard enhancements in Oracle Database 10g Release 1 and Release 2 that provide significant SQL Apply and logical standby improvements.

For an extensive list of the new features available in Oracle Database 10g Release 1, refer to the [Oracle Database New Features Guide](#)[5] for Release 1. For the new features in Oracle Database 10g Release 2, refer to [Oracle Database New Features Guide](#)[6] for Release 2.

### Oracle Database 10g Release 1 Enhancements

The following sections describe the SQL Apply and logical standby enhancements introduced in Oracle Database 10g Release 1.

#### Standby Redo Log Files

Logical standby databases can now use standby redo log files. Standby redo log files enable Data Guard to be configured for all modes of data protection, including those that guarantee no data loss. For more information about standby redo log files, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

#### Real-Time Apply

SQL Apply supports real-time apply when standby redo log files are present. With real-time apply, SQL Apply can apply redo data as it is received, without waiting for the current standby redo log file to be archived. For more information about real-time apply, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

#### Support for Maximum Protection Mode

Logical standby databases using SQL Apply can be configured for the highest possible level of data protection using Data Guard Maximum Protection mode. For

more information about protection modes, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

### **Zero Downtime Instantiation**

You can create a logical standby database without shutting down or quiescing the primary database. Zero downtime is achieved by using an online backup of the primary database and creating a logical standby control file.

### **Improved SQL Apply Performance**

Oracle Database 10g Release 1 SQL Apply performance is 25% better (as measured by the standby apply rate) than what could be achieved by a comparable configuration using Oracle9i Database Release 2.

### **Faster SQL Apply Switchover and Failover**

Oracle Database 10g reduces the time for a switchover to complete by introducing a prepare phase that executes before the actual outage occurs. A switchover to a logical standby database can complete in less than 30 seconds in Oracle Database 10g. For more information about switchover and failover, refer to the [“Switchover and Failover Best Practices: Oracle Data Guard 10g Release 2”](#)[8] white paper.

### **Identification of Unnecessary Log Files**

Oracle Database 10g introduces the `DBMS_LOGSTDBY.PURGE_SESSION` PL/SQL procedure. This procedure identifies the archived log files that have been applied to the standby database and are no longer required by SQL Apply. Once identified, you can issue operating system commands to delete some or all of the unnecessary archived redo log files.

### **Integration with Flashback Database to Resolve Logical Failures**

When configured on an Oracle Database 10g logical standby database, Oracle Flashback Database can recover failures to the logical standby database more efficiently by “rewinding” the standby database to the time before the failure occurred.

Flashback Database also removes the need to re-create the primary database after a failover. After a failover, the original primary database can now be flashed back to a point in time before the failover and converted into a standby database. Data Guard automatically applies all of the logs generated since the failover, resynchronizing the original primary (new standby) with the new primary. This process quickly restores high availability and data protection. Once this process is complete, the original primary database can be switched back to the primary role, returning the configuration to its original state prior to failover.

For additional information on Flashback Database, refer to the [Oracle Database Backup and Recovery Basics](#)[9] manual.

### **Support for More Data Types**

SQL Apply in Oracle Database 10g Release 1 supports more data types than past releases. Support has been added for index-organized tables without LOBs or overflow segments, and LONG, LONG RAW, and NCLOB columns. For a complete list of supported and unsupported data types in Oracle Database 10g Release 1, refer to the [Oracle Data Guard Concepts and Administration](#)[10] manual for Release 1.

### **Ability to Upgrade the Oracle Database Using SQL Apply**

Starting with Oracle Database 10g Release 1 Patch Set 1 (Oracle Database Release 10.1.0.3), you can use SQL Apply to upgrade the Oracle Database software. For additional information, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

### **Improved Redo Data Transmission Security**

Oracle Database 10g Release 1 improves the security of redo transmitted between the source and target databases with support for data encryption. For additional information, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

### **Oracle Database 10g Release 2 Enhancements**

The following sections describe the SQL Apply and logical standby enhancements introduced in Oracle Database 10g Release 2.

#### **Improved SQL Apply Performance**

Further internal optimizations have resulted in up to 18% improvement in SQL Apply over Oracle Database 10g Release 1.

#### **Support for Fast-Start Failover**

Fast-Start Failover provides the ability to automatically, quickly, and reliably fail over to a designated, synchronized standby database in the event of the loss of the primary database, without requiring that you perform complex manual steps to invoke the failover. For additional information on Fast-Start Failover, refer to the [Oracle Data Guard Broker](#)[11] manual and the [“Fast-Start Failover Best Practices: Oracle Data Guard 10g Release 2”](#)[15] white paper.

#### **Enhanced Data Guard Statistics**

Oracle Enterprise Manager now provides a visual interface to key Data Guard statistics that assist in managing and monitoring a logical standby database.

#### **Log Writer Process Asynchronous Transport Enhancements**

Asynchronous redo transmission using the log writer process (LGWR ASYNC) is optimized to further reduce overhead on the primary database.

### Modified SQL Apply Parameters for Improved Manageability

The LOG\_AUTO\_DELETE parameter in the DBMS\_LOGSTDBY.APPLY\_SET procedure controls whether SQL Apply deletes archived redo log files automatically when they are no longer needed. Also, the TRANSACTION\_CONSISTENCY parameter is renamed to PRESERVE\_COMMIT\_ORDER in this procedure. For additional information about the APPLY\_SET procedure, refer to [Oracle Database PL/SQL Packages and Types Reference](#)[12].

### Simpler Logical Standby Configuration with Less Tuning

Creation of a logical standby database no longer requires the creation of a specialized logical standby control file. You can now create a logical standby database easily from a physical standby database. This ability reduces the specialized manual operations for creating a logical standby database and improves Data Guard manageability. For more information about creating a logical standby database, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

### Support for More Data Types

In Oracle Database 10g Release 2, SQL Apply supports mining of redo records generated by changes to index-organized tables containing LOB columns and overflow segments. For a complete list of supported and unsupported data types in Oracle Database 10g Release 2, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual for Release 2.

## SQL APPLY BEST PRACTICES

The following sections describe best practices for configuring SQL Apply and logical standby database.

- Confirm Data Type Support
- Use Standby Redo Log Files
- Use Real-Time Apply
- Consider Using a No Data Loss Protection Mode
- Tune SQL Apply
- Benefit From Improved SQL Apply Performance
- Implement Faster SQL Apply Switchover and Failover
- Manage Logical Standby Archived Log Files
- Use Fast-Start Failover
- Use Enhanced Data Guard Management Statistics
- Use the Asynchronous Redo Transport Enhancements

These best practices were derived from extensive testing on Oracle Database 10g Release 2 as part of ongoing [MAA\[2\]](#) performance studies.

### Confirm Data Type Support

Before using SQL Apply, ensure that it can support the data types and tables from the primary database that you wish to maintain in your standby database. Refer to Section 4, “Prerequisite Conditions for Creating a Logical Standby Database”, and to Appendix C, “Data Type and DDL Support on a Logical Standby Database,” in the [Oracle Data Guard Concepts and Administration\[7\]](#) manual for Oracle Database Release 2. Note that any table that contains an unsupported data type is skipped in its entirety. Note also, that any user table placed within a schema listed as an internal schema in DBA\_LOGSTDBY\_SKIP will be skipped as these schemas are skipped in their entirety. In general, user data should not be stored in any internal schema supplied by Oracle.

One more query must be issued in addition to those referenced (above) in the Oracle Data Guard Concepts and Administration manual to identify all database objects that cannot be maintained by SQL Apply.

In Oracle Database 10g Release 1 and Release 2, the DBA\_LOGSTDBY\_UNUNSUPPORTED view does not return tables for certain unsupported types, such as those containing advanced queue tables or tables with only LONG data types. To completely identify all unsupported types run the following query

```
col owner format a9
col data_type format A35
col table_name format A35
(select COLS.owner, COLS.table_name, COLS.column_name,
COLS.data_type
from dba_tab_cols COLS, dba_tables TABLES
where COLS.owner not in
(select OWNER from DBA_LOGSTDBY_SKIP where
STATEMENT_OPT='INTERNAL SCHEMA')
and data_type not in
('BINARY_DOUBLE', 'BINARY_FLOAT', 'FLOAT', 'BLOB',
'CLOB', 'CHAR', 'DATE', 'LONG', 'LONG RAW', 'NCHAR',
'NCLOB', 'NUMBER', 'NVARCHAR2', 'RAW', 'VARCHAR', 'VARCHAR2')
and data_type not like 'TIMESTAMP%'
and data_type not like 'INTERVAL%'
and (COLS.owner=TABLES.owner
and COLS.table_name=TABLES.table_name)
and TABLES.secondary != 'Y'
union
select COLS.owner, COLS.table_name, COLS.column_name,
COLS.data_type
from DBA_LOGSTDBY_UNUNSUPPORTED COLS)
order by 1,2,3;
```

Note that any table containing user-defined data types starting with TIMESTAMP or INTERVAL will not be detected by this query even though they are

unsupported, and as with any table that includes user-defined data types, will be skipped in their entirety. To avoid this problem, Oracle recommends that user-defined data types follow naming conventions that are different from those reserved for data types provided by Oracle. This query is only necessary for RDBMS 10.1 releases prior to 10.1.0.6 and RDBMS 10.2 releases prior to 10.2.0.4. Later releases can use the `DBA_LOGSTDBY_UNSUPPORTED` view.

It should also be noted that both this query and the `DBA_LOGSTDBY_UNSUPPORTED` view do not take into account the database `COMPATIBLE` parameter setting. The `COMPATIBLE` setting can also restrict what is supported by the logical standby database. If you have `COMPATIBLE` set to a version lower than the version you have installed and running then you should consult the Data Guard Concepts and Administration guide for the release that you have the `COMPATIBLE` parameter set to for confirming supported data types.

### Use Standby Redo Log Files

With Oracle Database 10g, logical standby databases can use standby redo log files. Without the presence of standby redo log files, SQL Apply must wait for an archived log file to be received and registered on the standby database before SQL Apply can mine the log. Additionally, standby redo log files are required for all no data loss protection modes.

For more information about creating a logical standby database, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

**Note:** The size of the standby redo log files must be the same as the size of the online redo log files on the primary database.

### Use Real-Time Apply

With real-time apply, SQL Apply can apply redo data as it is received, without waiting for the current standby redo log file to be archived. Real-time apply requires standby redo log files.

To enable real-time apply, start SQL Apply with the following statement on the logical standby database:

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

### Consider Using a No Data Loss Protection Mode

If no data loss is required to meet recovery point objectives, then consider using either Maximum Protection or Maximum Availability protection mode. Both protection modes require standby redo log files. For more information about using a no data loss protection mode, including the network requirements, refer to the [“Primary Site and Network Configuration Best Practices”](#)[13] white paper and the [Oracle Data Guard Concepts and Administration](#)[7] manual.

## Tune SQL Apply

The following sections describe significant SQL Apply parameters that affect standby database performance and how to tune SQL Apply and logical standby database if default settings do not achieve the required apply rate.

### Adjust How Transactions Are Applied on the Logical Standby Database

With Oracle Database 10g, there are two significant SQL Apply parameters specific to tuning logical standby processes: `MAX_SERVERS` and `TRANSACTION_CONSISTENCY` (or `PRESERVE_COMMIT_ORDER`). These parameters are set using the `DBMS_LOGSTDBY.APPLY_SET` procedure and are described in the following sections.

#### **MAX\_SERVERS**

The SQL Apply parameter `MAX_SERVERS` controls the number of processes that the SQL Apply process can use. The default value in Data Guard 10g Release 2 is 9. An internal algorithm is used to allocate these processes to SQL Apply reader, preparer, builder, and applier processes. For a more complete description of these processes refer to Section 9.1 “Overview of the SQL Apply Architecture” in the [Oracle Data Guard Concepts and Administration](#)[7] manual. For optimum performance, `MAX_SERVERS` should be set to the higher of the following two values.

- `MAX_SERVERS` value based on the number of CPUs.

The recommended initial setting for this parameter remains unchanged from Oracle9i Database Release 2. It should be set to  $(3 * \text{CPU}) + 3$ .

```
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_SET ('MAX_SERVERS' , nn) ;
```

**Note:** where multi-core CPUs are utilized, use the total number of cores, not the number of CPU’s for the formula above. This means that in a 2 processor system with dual-core CPUs, the above formula would result in the following value for `MAX_SERVERS`:  $(3 * (2*2)) + 3 = 15$ .

- `MAX_SERVERS` value based on maximum transaction concurrency.

A second consideration for setting the optimum value for the `MAX_SERVERS` parameter involves the maximum transaction concurrency of the primary database. Maximum transaction concurrency is reported in the primary database AWR report under the “Undo Segment Stats” section of the “Undo Segment Summary”.

The following is an example of this AWR report for the second option:

```
Undo Segment Stats                                DB/Inst: TPCC/tpccp  Snaps: 15-20
-> Most recent 35 Undostat rows, ordered by Time desc
      Num Undo   Number of Max Qry  Max Tx  Tun Ret STO/   uS/uR/uU/
End Time      Blocks Transactions Len (s)  Concy   (mins) OOS    eS/eR/eU
```

```

-----
20-Oct 04:31 39,876      94,354      51      36      28 0/0  0/0/0/0/0/0
20-Oct 04:21 39,963      94,790      48      35      28 0/0  0/0/0/0/0/0
20-Oct 04:11 40,810      96,250      45      35      27 0/0  0/0/0/0/0/0
20-Oct 04:01 41,782      98,450      44      35      26 0/0  0/0/0/0/0/0
20-Oct 03:51 42,485     100,806      43      34      26 0/0  0/0/0/0/0/0
20-Oct 03:41 43,033     102,024      43      35      26 0/0  0/0/0/0/0/0
20-Oct 03:31 43,162     101,763      41      34      26 0/0  0/0/0/0/0/0
20-Oct 03:21 43,477     102,525      40      34      27 0/0  0/0/0/0/0/0
20-Oct 03:11 42,421     100,362      39      35      27 0/0  0/0/0/0/0/0
-----

```

The value for maximum transaction concurrency is the highest number of concurrent transactions seen during the two snapshots contained in an AWR report. In the report above, the maximum transaction concurrency is 36 transactions (the first row of the report). The formula for determining MAX\_SERVERS based upon transaction concurrency is the following:

$$(3 * \text{max transaction concurrency}) / 2$$

**Note:** When configuring MAX\_SERVERS, make sure there are sufficient CPU and I/O resources available to accommodate the number of processes configured and the increased CPU and I/O processing that will result. Refer to the “Finding System CPU Utilization” section in the [Oracle Database Performance Tuning Guide](#)[14].

Additionally, the SQL Apply MAX\_SERVERS parameter is dependent upon the PARALLEL\_MAX\_SERVERS initialization parameter. Therefore, you should set the PARALLEL\_MAX\_SERVERS initialization parameter to the value specified for the MAX\_SERVERS parameter, at a minimum.

```
SQL> ALTER SYSTEM SET PARALLEL_MAX_SERVERS=nn;
```

**TRANSACTION\_CONSISTENCY/PRESERVE\_COMMIT\_ORDER**

Oracle Database 10g Release 2 renames the TRANSACTION\_CONSISTENCY SQL Apply parameter to PRESERVE\_COMMIT\_ORDER for improved manageability.

SQL Apply in Oracle Database 10g supports two levels of transactional consistency. By default, transactions are applied in the exact order that they were committed on the primary database:

- In Oracle Database 10g Release 1, the default setting is TRANSACTION\_CONSISTENCY to FULL.
- In Oracle Database 10g Release 2, the default setting is PRESERVE\_COMMIT\_ORDER to TRUE.

SQL Apply uses this as the default mode because reporting applications often require strict transaction ordering between the primary and standby databases.

Optionally, for increased performance, transactions can be applied in a different order than on the primary database, but transaction consistency within each transaction is always maintained after switchover or failover. Following a role

transition, the new primary will be consistent and will contain all transactions up to a specific time or SCN. If this is acceptable for your application and increased performance is your goal, then consider the following change:

- In Oracle Database 10g Release 1, set `TRANSACTION_CONSISTENCY` to `NONE`.
- In Oracle Database 10g Release 2, set `PRESERVE_COMMIT_ORDER` to `FALSE`.
- MAA tests have shown there can be a 40-60% apply performance improvement in this mode.

### **Use the Standard Oracle Tuning Methodology**

To optimize the performance of a SQL Apply configuration, it is important to understand how the workload to maintain a logical standby database is different than the original workload executed on the primary database.

For example, the SQL statements executed at the primary database are distinctly different than those executed at the logical standby database. In an OLTP application, the primary database probably has many simple queries that can be shared, DML statements (including array inserts), and DDL statements. Because the logical standby is often used for both reporting and disaster recovery, a typical logical standby is likely to have more complex queries used for reporting and simple DML and DDL statements that are reconstructed after mining and analyzing the redo. The SQL Apply processes break down array inserts and batch updates into simpler insert, update, and delete statements. Furthermore, the SQL Apply processes use the shared pool for the logical change record (LCR) cache, which is a region of the shared pool used exclusively by SQL Apply processes.

These differences can create different memory requirements for the shared pool and library cache between the primary and standby databases. Similarly, you may have to tune other memory structures, such as the buffer cache, independently to further enhance SQL Apply performance.

With these facts in mind, the tuning of a logical standby database begins by following standard Oracle procedures as described in the [Oracle Database Performance Tuning Guide](#)[14]. Automatic Workload Repository (AWR) reports should be generated over relevant periods of time, and the resulting information should be used to tune the database, such as analyzing the most common wait events, latch contention, and system statistics. Refer to the [Oracle Database Performance Tuning Guide](#)[14] for Oracle Database 10g Release 2 for more information on assessing performance and identifying the top wait events.

### Additional Tuning Guidelines Specific to Logical Standby Databases

More information on SQL Apply tuning is provided in Section 9.6, “Tuning a Logical Standby Database” in the [Oracle Data Guard Concepts and Administration](#)[7] manual. Guidance is provided to complete the following tasks:

- Create primary key RELY constraints to enhance performance.
- Gather statistics for the cost-based optimizer to determine optimal query execution path.
- Adjust the number of SQL Apply processes allocated to preparers and appliers to enhance throughput
- Adjust the memory for LCR Cache to achieve maximum throughput.

If these additional steps are taken and performance goals are not yet realized, please refer to information provided in MetaLink Note [312434.1](#)[3], “Oracle 10g Data Guard SQL Apply Troubleshooting” for additional guidance.

### Benefit From Improved SQL Apply Performance

The best practices described above when combined with enhancements in Oracle Data Guard 10g, can yield a significant increase in SQL Apply performance compared to Oracle9i Data Guard. This section discusses SQL Apply performance for a given workload scenario. In this scenario, a workload consisting of four distinct types of transactions were run against the primary database:

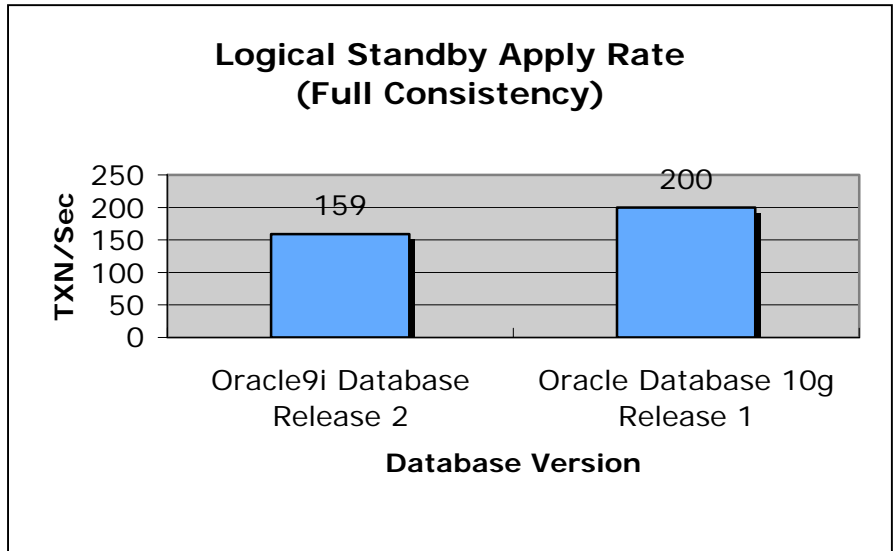
- Approximately 44% of all transactions modified data in six different tables, with a mixture of DML operations that manipulated approximately 25 rows.
- Approximately 44% of all transactions updated information in three different tables and inserted data into a fourth table.
- Approximately 4% of all transactions modified approximately 150 rows over four distinct tables.
- Approximately 8% of all transactions were split between two complex query transactions.

All transactions used either a primary or unique index in their access plans. No DDL transactions were performed in the transaction mix. The logical standby database was configured using Oracle best practices. Unless otherwise noted, MAX\_SERVERS was set at the best practice value based on the number of CPUs in each test configuration and the default setting for transaction consistency was used, applying transactions to the standby database in the exact order in which they were committed on the primary.

Test results were as follows:

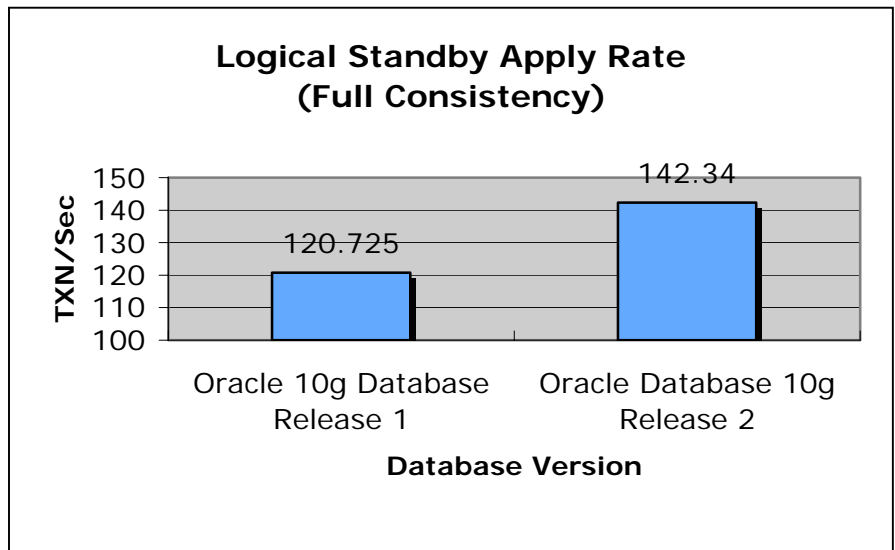
- Oracle Data Guard 10g Release 1 achieved 25% greater SQL Apply performance than using Oracle9i (Figure 1). The test system was an 8 CPU HP Server running HP-UX 11i with 10GB of memory allocated to Oracle.

Figure 1: SQL Apply Performance Increase SQL Apply 9i vs 10gR1



- Separate testing using Oracle Data Guard 10g Release 2 showed an additional 18% performance increase over Oracle Database 10g Release 1 (see Figure 2). The test configuration was an Intel 2 CPU Sun Server with Hyper Threading enabled running RedHat 3 (32Bit) with 2.5GB of memory allocated to Oracle.

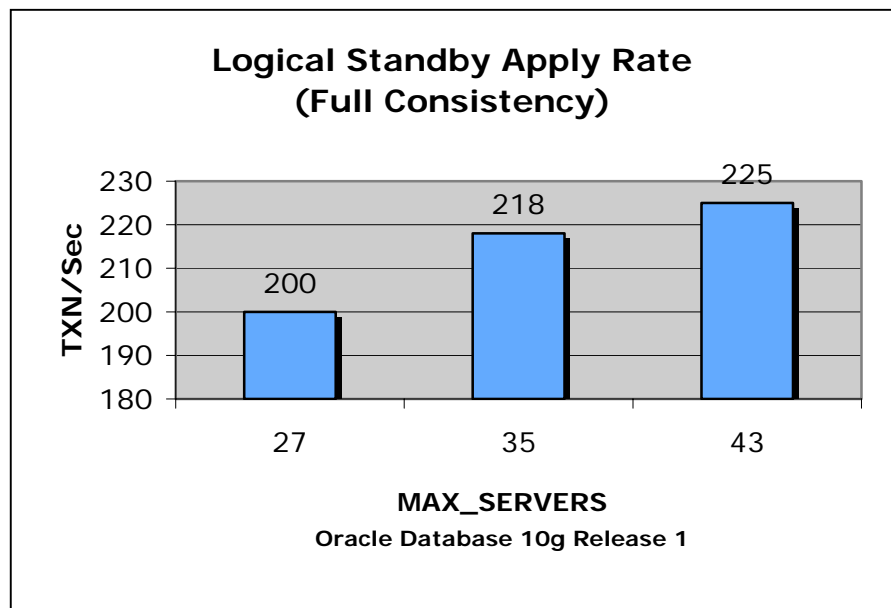
Figure 2: SQL Apply Performance Increase: SQL Apply 10gR1 vs 10gR2



**Performance Improvement: Increasing MAX\_SERVERS**

It may be possible to further enhance SQL Apply performance by increasing the value for MAX\_SERVERS (Figure 3). The initial MAX\_SERVERS value of 27 used in the Data Guard 10g Release 1 testing was the “best practice value” based upon the number of CPUs in the test configuration. Subsequent test runs using higher MAX\_SERVERS values demonstrated there was available memory and CPU capacity to achieve another 12% increase in performance. Care must be taken, however, when exceeding the best practice recommendation for MAX\_SERVERS to avoid a situation where SQL Apply begins competing for system resources needed by other applications that may be running on the same system.

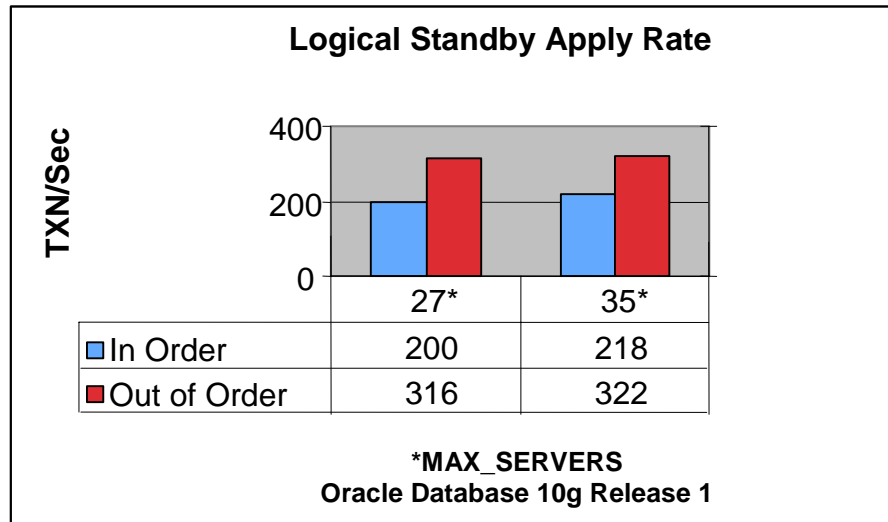
**Figure 3: Performance Improvement: Increasing MAX\_SERVERS**



**Performance Improvement: Changing COMMIT\_ORDER**

As previously noted, SQL Apply in Oracle Database 10g supports two modes of consistency: transactions may be applied in order or out of order. When requirements allow transactions to be applied out of order, increased SQL Apply performance can be achieved by changing the default setting of the TRANSACTION\_CONSISTENCY to NONE (Data Guard 10g Release 1) or PRESERVE\_COMMIT\_ORDER parameter to FALSE (Data Guard 10g Release 2). MAA tests demonstrated that a 58% and 48% performance improvement could be achieved when SQL Apply was configured to apply transactions out of order with MAX\_SERVERS set to 27 and 35 (Figure 4).

Figure 4: Performance Improvement: Changing COMMIT\_ORDER



### Implement Faster SQL Apply Switchover and Failover

Switchover operations can be completed in less than one minute, and failover operations can be completed in 15-30 seconds.

For best practices and tips to achieve the optimal switchover and failover, refer to the MAA white paper [“Switchover and Failover Best Practices: Oracle Data Guard 10g Release 2”](#)[8].

### Flash Recovery Area and Standby Archived Log Files

In Oracle Database 10g, it is not possible for a Logical Standby database to use the Flash Recovery Area for storing archived log files that result from redo shipped by the primary database. These archived log files must be archived via a second log\_archive\_dest\_n parameter defined on the standby database. For example:

```
SQL> alter system set log_archive_dest_n =
        'location=+FLASH/boston/archivelog/srl/
        valid_for=(standby_logfiles,standby_role)'
```

If you use the same physical device as the Flash Recovery Area, then the archived log files that result from redo shipped by the primary database will compete for space on that device with the archive logs and other recovery files generated by the Logical Standby database. This can lead to the Flash Recovery Area potentially not having enough space to keep the Logical Standby database running as it is not able to manage files outside the Flash Recovery Area. To alleviate this possibility, please refer to *Manage Logical Standby Archived Log Files* below.

## Manage Logical Standby Archived Log Files

Oracle Database 10g Release 2 introduces a new feature that automatically purges standby archived log files that are no longer required by SQL Apply. This is achieved through the parameter `LOG_AUTO_DELETE` in `DBMS_LOGSTDBY.APPLY_SET` procedure, which is enabled when SQL Apply starts. Note that you can disable automatic log file management by setting this parameter to `FALSE`. If so disabled, log files will need to be managed as they are for Oracle Database 10g Release 1 logical standby databases, described below.

Oracle Database 10g Release 1 requires that the logical standby's archived log files be purged manually. However, Oracle Database 10g introduces a PL/SQL procedure that cleans up the metadata in the logical standby database and generates a report of the archived log files that have been applied and are no longer required by SQL Apply.

To purge the logical standby session of metadata that is no longer needed, run the following PL/SQL statement:

```
SQL> EXECUTE DBMS_LOGSTDBY.PURGE_SESSION;
```

This statement also updates the `DBA_LOGMNR_PURGED_LOG` data dictionary view. This view displays the archived redo log files that are no longer needed by SQL Apply.

Query the `DBA_LOGMNR_PURGED_LOG` view to list the archived redo log files that can be removed. For example:

```
SQL> SELECT * FROM DBA_LOGMNR_PURGED_LOG;
```

```
FILE_NAME
-----
/boston/arc_dest/arc_1_40_509538672.log
/boston/arc_dest/arc_1_41_509538672.log
/boston/arc_dest/arc_1_42_509538672.log
/boston/arc_dest/arc_1_43_509538672.log
/boston/arc_dest/arc_1_44_509538672.log
/boston/arc_dest/arc_1_45_509538672.log
/boston/arc_dest/arc_1_46_509538672.log
/boston/arc_dest/arc_1_47_509538672.log
```

Use an operating system command to delete the archived redo log files listed by the query.

For additional information, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

## Use Fast-Start Failover

You can configure a logical standby database as the target of a fast-start failover. Fast-Start Failover is an Oracle Data Guard 10g Release 2 feature that automatically, quickly, and reliably fails over to a designated, synchronized standby database in the

event of loss of the primary database, without requiring manual intervention to execute the failover.

In addition, following a fast-start failover, the original primary database is automatically reconfigured as a new standby database upon reconnection to the configuration. This ability enables Data Guard to restore disaster protection in the configuration quickly and easily, returning the database to a protected state as soon as possible.

Fast-Start Failover requires use of the Data Guard broker and for the Data Guard configuration to be set to Maximum Availability mode with standby redo log files and Flashback Database enabled.

For more information about using Fast-Start Failover with SQL Apply, refer to the [“Fast-Start Failover Best Practices: Oracle Data Guard 10g Release 2”](#)[15] white paper:

### **Use Enhanced Data Guard Management Statistics**

Beginning with Oracle Database 10g Release 2, Oracle Enterprise Manager provides a visual interface to key Data Guard statistics, including:

- Estimated Failover Time (seconds)
- Apply Lag (seconds)
- Redo Generation Rate (KB/second)
- Redo Apply Rate (KB/second)
- Transport Lag (seconds)
- Data Guard Status
- Fast-Start Failover Occurred / SCN and Time

For more information on these statistics, refer to the [Oracle Data Guard Concepts and Administration](#)[7] manual.

### **Use the Asynchronous Redo Transport Enhancements**

In Oracle Database 10g Release 2, redo transport services have been significantly enhanced, further reducing primary system overhead, improving network throughput, and making it possible to achieve even greater geographic separation (and disaster protection) between data centers for applications with high workload.

When you use asynchronous redo transmission with the log writer process (LGWR ASYNC), the log writer process writes to the local online redo log file, while the network server processes (LNS<sub>n</sub>, one for each destination) asynchronously transmit the redo to remote destinations. The LGWR ASYNC process continues processing the next request without waiting for the LNS network I/O to complete. For more information about LGWR ASYNC, refer to the [Oracle Data Guard Concepts and](#)

[Administration](#)[7] manual and the 2005 Oracle Open World presentation "[What They Didn't Print in the Doc: HA Best Practices by Gurus from Oracle's Maximum Availability Architecture Team](#)"[16].

Please refer to the [MAA](#)[2] Web page for an upcoming Data Guard 10g Release 2 white paper on this subject.

## **CONCLUSION**

Oracle Database 10g enhances the features and performance of Data Guard SQL Apply. SQL Apply with logical standby database is a viable option for customers who need to implement a disaster recovery solution and use the same resources for reporting and decision support operations.

## APPENDIX A

### Quick Reference - Upgrading from Oracle9i Database to Oracle Database 10g

The following list highlights the significant new features to consider when upgrading from Oracle9i Database Release 2 to Oracle Database 10g:

- Real-time Apply enables real-time reporting against the standby database and reduces the time required for switchover or failover to the standby database.
- Oracle Database 10g removes the need to use undocumented parameters such as `_EAGER_SIZE` in all but rare circumstances (refer to MetaLink Note [312434.1](#)[3], “Oracle 10g Data Guard SQL Apply Troubleshooting”).

To remove the `_EAGER_SIZE` parameter, run the following procedure:

```
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_UNSET ('_EAGER_SIZE');
```

- Oracle Database 10g removes the `READ_ONLY` transaction consistency setting.
  - To change the transaction consistency setting in Oracle Database 10g Release 1, run the following procedure:

```
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_SET ('TRANSACTION_CONSISTENCY', 'FULL');
```
  - To change the transaction consistency setting in Oracle Database 10g Release 2, run the following procedure:

```
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_SET ('PRESERVE_COMMIT_ORDER', TRUE);
```

For additional information about upgrading from Oracle9i Database Release 2 to Oracle Database 10g, refer to MetaLink Note [278108.1](#)[17], “Upgrading to Oracle 10g with a Logical Standby in Place.”

## REFERENCES

1. Oracle Data Guard  
<http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html>
2. *Oracle Maximum Availability Architecture*  
<http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>
3. Metalink Note 312434.1: “Oracle 10g Data Guard SQL Apply Troubleshooting”  
[http://metalink.oracle.com/metalink/plsql/ml2\\_documents.showDocument?p\\_database\\_id=NOT&p\\_id=312434.1](http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=312434.1)

4. Metalink Note 274170.1: “Logical Standby Master Index Page”  
[https://metalink.oracle.com/metalink/plsql/ml2\\_documents.showDocument?p\\_database\\_id=NOT&p\\_id=274170.1](https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=274170.1)
5. *Oracle Database New Features Guide* for Oracle Database 10g Release 1  
[http://otn.oracle.com/pls/db10g/db10g.to\\_toc?partno=b10750](http://otn.oracle.com/pls/db10g/db10g.to_toc?partno=b10750)
6. *Oracle Database New Features Guide* for Oracle Database 10g Release 2  
[http://otn.oracle.com/pls/db102/db102.to\\_toc?partno=b14214](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14214)
7. *Oracle Data Guard Concepts and Administration* for Oracle Database 10g Release 2  
[http://otn.oracle.com/pls/db102/db102.to\\_toc?partno=b14239](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14239)
8. “Switchover and Failover Best Practices: Oracle Data Guard 10g Release 2”  
[http://www.oracle.com/technology/deploy/availability/pdf/MAA\\_WP\\_10gR2\\_SwitchoverFailoverBestPractices.pdf](http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_SwitchoverFailoverBestPractices.pdf)
9. *Oracle Database Backup and Recovery Basics*  
[http://otn.oracle.com/pls/db102/db102.to\\_toc?partno=b14192](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14192)
10. *Oracle Data Guard Concepts and Administration* for Oracle Database 10g Release 1  
[http://otn.oracle.com/pls/db10g/db10g.to\\_toc?partno=b10823](http://otn.oracle.com/pls/db10g/db10g.to_toc?partno=b10823)
11. *Oracle Data Guard Broker*  
[http://otn.oracle.com/pls/db102/db102.to\\_toc?partno=b14230](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14230)
12. *Oracle Database PL/SQL Packages and Types Reference*  
[http://otn.oracle.com/pls/db102/db102.to\\_xlink?xlink=ARPLS363](http://otn.oracle.com/pls/db102/db102.to_xlink?xlink=ARPLS363)
13. “Primary Site and Network Configuration Best Practices”  
[http://www.oracle.com/technology/deploy/availability/pdf/MAA\\_WP\\_10gR2\\_DataGuardNetworkBestPractices.pdf](http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_DataGuardNetworkBestPractices.pdf)
14. *Oracle Database Performance Tuning Guide*  
[http://otn.oracle.com/pls/db102/db102.to\\_toc?partno=b14211](http://otn.oracle.com/pls/db102/db102.to_toc?partno=b14211)
15. “Fast-Start Failover Best Practices Oracle Data Guard 10g Release 2”  
[http://www.oracle.com/technology/deploy/availability/pdf/MAA\\_WP\\_10gR2\\_FastStartFailoverBestPractices.pdf](http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_FastStartFailoverBestPractices.pdf)
16. “What They Didn't Print in the Doc: HA Best Practices by Gurus from Oracle’s Maximum Availability Architecture Team”  
[http://www.oracle.com/technology/deploy/availability/pdf/S936\\_ToMeeks.ppt.pdf](http://www.oracle.com/technology/deploy/availability/pdf/S936_ToMeeks.ppt.pdf)
17. Metalink Note 278108.1: “Upgrading to Oracle 10g with a Logical Standby in Place”  
[http://metalink.oracle.com/metalink/plsql/ml2\\_documents.showDocument?p\\_database\\_id=NOT&p\\_id=278108.1](http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=278108.1)



**SQL Apply Best Practices: Oracle Data Guard 10g**

April 2007

Author: Andrew Babb

Updated by Ray Dutcher

Contributing Authors: Lawrence To, Doug Utzig, Ashish Ray, Raymond Dutcher, Michael Smith, Randy Urbano, Joseph Meeks

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](http://oracle.com)

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.