

# Minimizing Downtime During Large Database Migrations with Oracle Data Guard

BY KAI YU  
DAVID BRYAN MAR

Migrating a large Oracle® database can require significant downtime, potentially disrupting critical enterprise systems. By integrating Oracle Data Guard and Oracle Recovery Manager features into the process, administrators can minimize this downtime and help maintain application availability during the migration.



**T**raditional methods of migrating large Oracle databases have typically required significant downtime over the course of days or even weeks to help ensure that the target database remains consistent with the source, potentially disrupting critical enterprise systems. Taking advantage of Oracle Data Guard and Oracle Recovery Manager (RMAN) features during a migration, however, can enable administrators to minimize this downtime while still keeping the target database up-to-date. This article outlines this process and describes an example migration performed by Dell engineers to demonstrate its key steps.

**Related Categories:**

Application servers

Database

Dell PowerEdge servers

Oracle

Visit [www.dell.com/powersolutions](http://www.dell.com/powersolutions) for the complete category index.

**Disadvantages of traditional migration methods**

Large Oracle database migrations are traditionally performed using one of three methods:

- **Export/import:** This method involves performing a full export of the source database to multiple dump files, copying these files to the target server, and then importing them into the target database. These steps can take several days, or even weeks, to complete with large databases—and during the migration, the target database cannot capture transactions against the source database.
- **Table copy:** This method involves exporting the database object structures from the source database and importing them into the target database. Administrators can copy table data using the SQL\*Plus copy utility or the

SQL\*Plus insert as select statement through database links. The table copy method is typically faster than the export/import method if administrators carry out multiple copy sessions in parallel among different Oracle Real Application Clusters (RAC) instances. Its drawback, as with the export/import method, is that the target database cannot capture transactions against the source database during the migration.

- **Transportable tablespace:** This method involves exporting the metadata, copying the data files from the source system to the target system, then importing the data files to the target database. Like the table copy method, this method is typically faster than the export/import method, but it also shares the same drawback that the target database cannot capture transactions against the source database during the migration.

Each of these methods typically requires administrators to shut down all transactions against the source database during the migration to help keep the target database up-to-date, which can result in significant downtime for large databases. However, because many critical applications must provide continuous services, even planned downtime for databases can be costly. To help avoid this problem, administrators can use a migration approach based on Oracle Data Guard designed to require downtime only when switching the application database connections to the target database.

“Following a system failure, the standby database can take over as primary database with the last transaction before failure intact.”

### Oracle Data Guard–based migration process

Oracle Data Guard is designed to help enterprises implement disaster recovery systems by maintaining copies (called standby databases) of production databases (called primary databases). Administrators can initially create a standby database from a backup of their primary database, after which Oracle Data Guard can automatically maintain the standby database as a transaction-consistent copy by transmitting and applying primary database redo logs. Following a system failure, the standby database can then take over as primary database with the last transaction before failure intact.

To help minimize downtime for the source database during a migration while keeping the target database up-to-date, administrators can set up the target database as a standby database of the source database and configure Oracle Data Guard to automatically transmit and apply redo logs to the target database. Because redo log application typically occurs more frequently than primary database transactions, at some point the target database should become in sync with the source database. Once this occurs, administrators can complete the migration process by shifting the target database into the primary role and moving the application database connections to the target database—the only step requiring database downtime, and one that administrators can typically complete quickly.

### System configuration for example Oracle Data Guard–based migration

To help demonstrate the Oracle Data Guard–based migration process, in September 2006 Dell engineers used it to migrate a live 1.6 TB Oracle production database to a target system. Figure 1 summarizes the configuration of each system. The source system was an Oracle RAC 10g cluster of two Dell™ PowerEdge™ 6650 servers using raw devices on EMC® Symmetrix® storage. The target system was an Oracle RAC 10g cluster of four Dell PowerEdge 6850 servers using Oracle Automatic Storage Management (ASM) disk groups on EMC Symmetrix storage. Because multiple critical applications were utilizing the source database, downtime during the migration could have had a significant impact.

On the target system, the total storage of 4 TB was divided in half, with 2 TB configured as database logical units (LUNs) on which ASM disk groups were built, and 2 TB configured as raw devices to provide temporary

storage for the initial standby database files. After the migration was complete, the Dell team reconfigured this temporary storage as database LUNs to help provide room for future database growth.

### Implementation of example Oracle Data Guard–based migration

Before carrying out the migration, the Dell team configured the four-node Oracle RAC 10g cluster that would serve as the target system, which included installing Oracle RAC 10g and creating the Oracle ASM instances and a seed database. The source and target systems both ran the same version of Oracle RAC 10g, which is necessary for a successful migration.

### Preparing the source database for migration

To prepare the source database for migration and help ensure that all transactions were recorded in the redo logs, the Dell team enabled the force logging mode on the source database by using the following SQL command:

```
alter database force logging;
```

The Dell team also created two additional undo tablespaces, undo3 and undo4, for nodes 3 and 4 along with a password file for the target database. These modifications did not affect source database operations.

### Configuring the target database as a standby database

To create the standby database on the target system, the Dell team created a set of raw devices on the temporary storage with the same raw device

	Source	Target
<b>Servers</b>	Two Dell PowerEdge 6650 servers	Four Dell PowerEdge 6850 servers
<b>Processors</b>	Four single-core Intel® Xeon® processors at 1.6 GHz	Four dual-core Intel Xeon processors at 3.66 GHz
<b>Memory</b>	8 GB	8 GB
<b>Storage</b>	1.7 TB	4 TB
<b>OS</b>	Red Hat® Enterprise Linux AS 3	Red Hat Enterprise Linux AS 3
<b>Database</b>	Oracle Database 10.1.03	Oracle Database 10.1.03
<b>Database storage</b>	Raw devices on EMC Symmetrix storage	Oracle ASM disk groups on EMC Symmetrix storage
<b>Database cluster</b>	Two-node Oracle RAC 10g cluster	Four-node Oracle RAC 10g cluster
<b>Network interface cards (NICs)</b>	Two embedded Broadcom Gigabit* Ethernet NICs	Broadcom BCM5704 dual Gigabit Ethernet controller

\* This term does not connote an actual operating speed of 1 Gbps. For high-speed transmission, connection to a Gigabit Ethernet server and network infrastructure is required.

Figure 1. Source and target system configuration for example Oracle Data Guard–based database migration

```
alter database force logging;
alter tablespace data1 begin backup;
!dd if=/dev/oradata/data1-1 ibs=1024k | ssh target_ip_address dd of =/dev/oradata/data1-1
!dd if=/dev/oradata/data1-2 ibs=1024k | ssh target_ip_address dd of =/dev/oradata/data1-2
alter tablespace data1 end backup;
```

**Figure 2.** SQL script for copying the raw devices of the data1 tablespace from the source system to the target system

```
alter system set log_archive_dest_2='SERVICE=target ARCH NOAFFIRM REOPEN=10 MAX_FAILURE=5 OPTIONAL'
scope=both sid='*';
alter system set fal_client=source scope=both sid='*';
alter system set fal_server=target scope=both sid='*';
alter system set log_archive_dest_state_2 = enable sid='*';
```

**Figure 3.** SQL commands to configure the Oracle Data Guard redo log destination and database initialization parameters

name, path, and size as those on the source system. They then created a hot backup of the source database by copying each raw device file from the source system to the corresponding raw device on the target system. The SQL script shown in Figure 2 provides an example of how to copy all the raw devices of the data1 tablespace from the source system to the temporary storage on the target system. The Dell team created a similar script for each tablespace and ran the copy operations in parallel across all nodes.

In addition to copying the raw device data files from the source database, the Dell team also created a standby control file from the source database and copied it to the target system with the following SQL command:

```
alter database create standby controlfile as
'/oracle/dbs/stbycf.ctl';
```

The Dell team also created a standby database parameter file, `initstandby.ora`, from the primary database and provided the standby control file name as the value for the `control_files` database initialization parameter:

```
*.control_files = '/oracle/dbs/stbycf.ctl'
```

Next, the Dell team started the standby database on node 1 of the target system as `nomount`, and later mounted it as the standby database on node 1 using the following SQL commands:

```
startup nomount pfile=initstandby.ora;
alter database mount standby database;
```

### Configuring the automatic redo log transmission and its application

Oracle Data Guard uses archive processes or the log writer process to collect redo log data and transmit it to standby databases. The Dell team

configured a new archive process (archive process 2) in the source database to transmit the redo logs to the target system by pointing the log destination to the target database and setting the `fal_client` and `fal_server` database initialization parameters using the SQL commands shown in Figure 3. In these commands, `source` and `target` are the entries for the source and target databases configured in the `$TNS_ADMIN/tnsnames.ora` file on the source system.

The Dell team next changed the target database to managed recovery mode, which allows archive redo logs to be transmitted from the source database and automatically applied to the target database, with the following SQL command:

```
alter database recover managed standby database
disconnect from session;
```

As the system copies the data files to the target system, the source system can still service transactions. These transactions generate redo logs, which then become backlogs of archive redo logs that must be applied to the target database before administrators can migrate it to Oracle ASM.

### Configuring the target database to use Oracle Automatic Storage Management

To take advantage of Oracle ASM, administrators must configure the target database to use this feature, a process that includes migrating the control file, `spfile`, database files, and online redo log files. To migrate the control file to ASM, the Dell team first changed the `control_files` database parameter on the target database to point to the ASM disk group location:

```
*.control_files = '+DATADG/target/controlfile'
```

They next used the following RMAN command to migrate the /oracle/dbs/stbycf.ctl control file to the ASM DATADG disk group:

```
restore controlfile from '/oracle/dbs/
stbycf.ctl';
```

To migrate the data files to the DATADG disk group, the Dell team executed the following RMAN commands for each tablespace on the target database:

```
backup as copy tablespace "DATA1" format
'+DATADG';
switch tablespace "DATA1" to copy;
```

*Note:* Migrating data files and applying redo logs cannot happen in parallel. Before migrating the data files, the Dell team took the target database out of managed recovery mode using the following SQL command:

```
alter database recover managed standby database
cancel;
```

Migrating all 1.6 TB of data files from raw devices to ASM took approximately 10 hours. To help ensure that the target database did not lag behind the source database while not in managed recovery mode, the Dell team switched the target database between managed recovery mode and ASM migration at regular intervals.

### Changing the target database to the primary role

Before the shifting the application database connections from the source database to the target database, administrators must change the target database from the physical standby role to the primary role. Because the target database can no longer accept redo logs from the source database once it is in the primary role, administrators must first ensure that the target database is in sync with the source database, then shut down the source database.

The Dell team carried out this process by archiving all redo logs and validating that the last redo log from the source database had been applied to the target database, then shutting down the source database by executing the following SQL commands on the source database.

```
alter system archive log all;
shutdown immediate;
```

Next, they checked for any additional redo logs generated before the source database shut down and manually applied these logs to the target database. Finally, they shifted the target database to the primary role and

opened it for application transactions by executing the following SQL commands on the target database:

```
alter database recover managed standby database
finish skip standby logfile;
alter database commit to switchover to primary;
```

### Switching application database connections to the target database

Before switching the application database connections to the target database, the Dell team first re-created all online redo logs in Oracle ASM on the target database using the following SQL commands:

```
alter database clear unarchived logfile group 1;
alter database drop logfile group 1;
alter database add logfile Thread 1 group 1
'+ DATADG';
```

After re-creating these logs, the Dell team started database instance 2, then switched the application database connections to the target database by using the new tnsnames.ora entries for the target database instances. Meanwhile, they also brought Oracle RAC instances 3 and 4 on nodes 3 and 4 online and joined them to the database cluster to help provide application load balancing.

### Oracle database migration with minimal downtime

Traditional methods of migrating large Oracle databases have typically required significant downtime, which can be unacceptable in enterprise environments and make system upgrades difficult to perform. By taking advantage of Oracle Data Guard and Oracle Recovery Manager features when migrating a large Oracle database, administrators can avoid these long periods of downtime and help maintain the availability of critical applications during the migration. [🔗](#)

*Kai Yu is a senior systems engineering consultant on the Dell Database and Applications Solutions engineering team specializing in Oracle RAC, database migration, and the Oracle E-Business Suite, and has previously been an Oracle database and applications database administrator. Kai has a B.E. and an M.E. in Computer Science from Huazhong University of Science and Technology and an M.S. in Computer Science from the University of Wyoming.*

*David Bryan Mar is a technical engineering manager on the Dell Oracle Database Solutions engineering team, where his product development responsibilities include Oracle RAC on the Microsoft® Windows® and Linux® operating systems and deployment methodologies. He has a B.S. in Computer Engineering and a master's degree in Computer Science from Texas A&M University.*