

the *Availability Digest*

www.availabilitydigest.com

Oracle Data Replication

September 2011

The key component for achieving high availability and continuous availability is data replication. It is the replication engine that maintains a remote copy of the production database so that it is protected against any disaster that might befall the production database.

Furthermore, if bidirectional replication is used, all nodes in the application network can participate in transaction processing. Should a node fail, all that needs to be done is to route further transactions to surviving nodes. Recovery times measured in seconds can be achieved. This is the active/active architecture.¹

By many measures, the Oracle database is the most widely used database today for critical corporate applications. Oracle has a rich set of data replication capabilities to support a range of availabilities and data protection. In this article, we look at these Oracle products. They include Oracle Data Guard, Oracle GoldenGate, and Oracle Streams.

A Product Summary

Oracle has a suite of data replication products. By and large, they are quite similar. They all read changes to a source database from the Oracle redo log and send the changes to one or more remote systems. At the remote systems, they can be applied to target databases or can be sent to applications as events.

Their newest replication engine is GoldenGate. Oracle acquired GoldenGate Software, Inc., in 2009. Indications are that GoldenGate is to be the preferred replication engine in the future. In fact, at Oracle World 2010, Oracle announced that Oracle Streams will continue to be supported but not be actively enhanced.² Oracle GoldenGate will be the future for Oracle replication.

The one capability that GoldenGate does not possess is synchronous replication. This need is filled by Oracle Data Guard. Oracle Data Guard can also provide replication between RAC (Real Application Cluster) clusters with automated switchover in the event of a production-node failure.³

A major feature of GoldenGate is its heterogeneity. Industry wisdom has it that Oracle's primary interest in GoldenGate is to use it to integrate Oracle into non-Oracle environments to become the database of record.

¹ *What is Active/Active?*, *Availability Digest*, October, 2006.

http://www.availabilitydigest.com/public_articles/0101/what_is_active-active.pdf

² *Oracle GoldenGate and Oracle Streams: The Future of Oracle Replication and Data Integration*, *Oracle presentation*.

<http://www.oracle.com/technetwork/database/features/availability/312833-129009.pdf>

³ *Oracle Active Data Guard and Oracle GoldenGate*, *Oracle white paper*.

<http://www.oracle.com/technetwork/database/features/availability/dataguardgoldengate-096557.html>

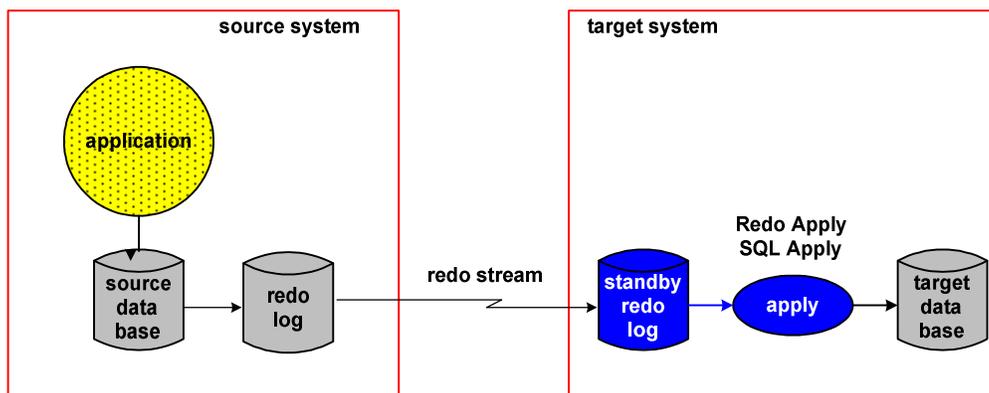
GoldenGate grew up in the HP NonStop world, where it was heavily used for data replication. Its bidirectional capabilities are used in many NonStop active/active applications. However, it seems that Oracle is turning its back on the NonStop community based on the significant license fee increases that it is exacting from current NonStop customers. Many of these customers are migrating to other NonStop replication engines such as Shadowbase from Gravic, Inc., (www.gravic.com/shadowbase) and DRNet from Network Technologies International (<http://network-tech.com/drnet.html>).

Oracle Data Guard

Oracle DataGuard provides asynchronous or synchronous replication services to create and manage one or more standby databases that are copies of a production database.⁴

Architecture

The Oracle database maintains a *redo log* that records every change made to the production database. Data Guard accomplishes data replication by shipping the production redo log to each standby database. At each standby database, Oracle applies the redo data to keep the standby database synchronized with the production database.



Oracle Data Guard Data Replication

Data Guard supports heterogeneous configurations. A standby database may be the same as the production database. In this case, redo changes are applied with Oracle's Redo Apply facility. On the other hand, the standby databases may be different. In this case, SQL Apply is used to transform the data in the redo log to standard SQL statements and then to execute these statements on the standby database.

Configuration

Data Guard can support up to nine standby databases. The standby databases are consistent and correct copies of the production database. As such, they can support many functions such as query, reporting, and backups. Furthermore, Data Guard can switch any standby database to production use to recover from a primary database failure.

The databases in a Data Guard configuration may be dispersed geographically so long as they can communicate with each other. This is important for disaster recovery and can be useful to position database copies close to communities of users.

⁴ Introduction to Oracle Data Guard, Oracle white paper.
http://download.oracle.com/docs/cd/B19306_01/server.102/b14239/concepts.htm

The primary and standby databases may each be managed by their own dedicated server, or they may each be databases in a RAC (Real Application Cluster).

If Data Guard should detect that a redo log on a standby system is corrupt, it will refresh that redo log from the production database or that of another standby system, if there is one.

Role Transitions

The production and standby roles of the databases in the network can be interchanged either via a switchover or a failover. A switchover is a controlled change of roles, usually done for planned maintenance. Switchover is done with no data loss.

A failover occurs following a catastrophic failure of the production database. One of the standby databases is configured to become the production database in the event of such a failure. In this case, some data that has not yet been replicated may be lost.

Should a standby database fail, the production database will queue further redo log blocks until the standby database is returned to service. At that time, the production database will drain its redo log queue to bring the standby database back into synchronization.

Role transitions are commanded manually. However, they can be simplified or automated using Oracle's Data Guard Broker.

Data Guard Broker

The Data Guard Broker is a distributed management facility that automates the creation, maintenance, and monitoring of a Data Guard configuration. It can be used via a command line interface or via a GUI interface provided by Oracle Enterprise Manager.

Data Guard Protection Modes

Data Guard provides three protection modes that offer various compromises between failure recovery time and data loss following a failure as well as performance during normal operation.

Maximum Protection Mode

The *maximum protection mode* ensures that no data will be lost following a production node failure. This mode uses synchronous replication. A transaction on the production system cannot complete until the redo log needed to recover that transaction has been written to at least one standby system.

If replication should fail for any reason, the production system is shut down until the problem can be corrected. This prevents a subsequent failure in the production system from causing the loss of data that has not been replicated.

Maximum Availability Mode

The *maximum availability mode* is similar to the maximum protection mode in that synchronous replication is used to keep the standby databases in synchronism with the production database. However, in this mode, the production system will continue to operate in the maximum performance mode described next if replication should fail. When replication is resumed, the redo log blocks that have not yet been replicated will be sent to the standby databases. When this change queue has been drained, synchronous replication resumes.

In this mode, no data loss will occur if replication fails. However, if the production system should fail before replication is resumed, the queued changes will be lost. Data loss requires a dual failure – a failure of the replication channel followed by a failure of the production system.

Maximum Performance Mode

Both the maximum protection mode and the maximum availability mode impose a performance penalty on the applications using the database. This is because each transaction is delayed until at least one standby has received all of the redo log blocks containing an operation for that transaction.

The *maximum performance mode* corrects this performance problem by using asynchronous replication. A transaction will commit on the production system immediately without waiting for a standby system. However, data loss following a production system failure is now possible. Any redo log entries that have not been replicated will be lost.

Other Features

Replication can be delayed by a specified time interval. This gives some protection against user errors which may erroneously modify or delete data. Alternatively, Data Guard's Flashback Database feature lets a database be restored to some previous point in time from the redo log without having to restore a backup and then roll transactions forward to the time of the error.

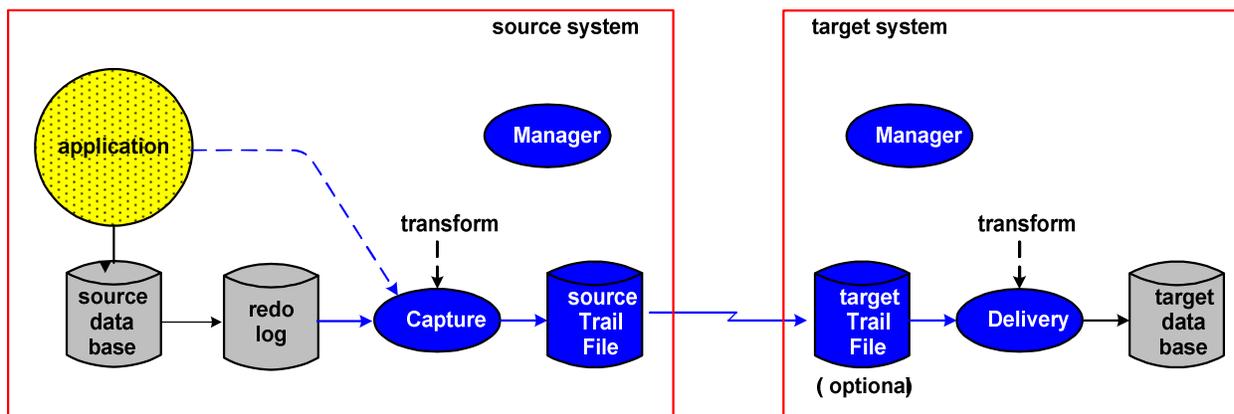
Because database synchronization is performed by replicating the redo log rather than actual database changes, corruption to the production database does not replicate to the standby databases. This is true so long as the redo log is not corrupted.

Oracle GoldenGate

The GoldenGate replication engine provides asynchronous replication between heterogeneous platforms, including Oracle.⁵ Furthermore, GoldenGate provides bidirectional replication supporting active/active configurations.

Architecture

The architecture of the GoldenGate replication engine is similar in many respects to Data Guard. GoldenGate follows a queue of changes made to the source database and sends them to the target database for applying to the target database.



GoldenGate Data Replication

⁵ Flexible Availability Options with GoldenGate's Transactional Data Management Platform (TDM), *Availability Digest*, February 2007. http://www.availabilitydigest.com/public_articles/0202/goldengate.pdf

GoldenGate provides the data transformation between the source and target databases to support heterogeneity, whether it be simply different data structures in otherwise identical databases or database systems from different vendors.

A Capture process receives changes to the source database. These changes could come from a variety of sources. They might be extracted from a transaction log (such as Oracle's redo log or the NonStop TMF audit trail) maintained by the source database or transaction manager. If such a log is not available, changes may be fed to the Capture process directly by the application or by an intercept library embedded in the application; or they may be fed by change information generated by database triggers.

Data changes captured by the Capture process are written to the source-side Trail File, which provides a queue of transactions to be sent to the target system. There may also be a target system Trail File which will queue changes prior to applying them to the target database. The Trail Files function as consistent en-route checkpoints for the data replication process so that replication can be continued without data loss following a fault, at least so far as the data that has made it to the Trail File is concerned,

The transfer of data changes from the source system to the target system is via IP (the Internet Protocol). The communication channel can be any medium that supports IP. There is no geographical limit on the separation of the source system from the target system.

Data changes are delivered to the Delivery process from the target-side Trail File if there is one or otherwise directly from the communication channel. The Delivery process uses the changes to update the target database.

The Manager controls the GoldenGate processes on its node via a command-line interface. It provides the facilities for configuration, critical event and threshold monitoring, log resource management, and the management and purging of old Trail Files.

GoldenGate supports dynamic rollback and reconstruction of a database for rapid recovery from errors that may have contaminated the database. Before-images are used to create reverse transactions to a specified point in time that is known to correspond to an accurate database. After-images are then used to create transactions to return the database to its proper current state.

Configuration

Flexible network technologies are supported. GoldenGate can be configured in a multinode peer-to-peer configuration (active/active). It can be configured to broadcast data changes from a master database to multiple slave databases or to consolidate changes made to several database copies onto a master database. It can support cascading of data down a multitiered database structure.

Transformation and Mapping

Database changes captured from the source database can be mapped and transformed into the formats required by the target database via GoldenGate-supplied functions, by user-supplied code installed as user exits, or by stored procedures in the database. Data mapping and transformation can be done either by the Capture process, by the Delivery process, or by both.

Data Collision Detection and Resolution

A data collision occurs under bidirectional replication when two users each attempt to modify the same row at the same time at two different nodes. GoldenGate can detect collisions and can be directed to take one of several actions, including:

- Ignore the collision.
- Retry the operation.
- Apply the increments to the field.
- Resolve with user-defined rules.
- Select a winning update based on the position of the source node in a nodal hierarchy.
- Suspend replication for external action.
- Log for manual resolution.

Zero Downtime Operations

GoldenGate can support upgrades to the system as well as migration to new system components without incurring planned downtime. Planned downtime for maintenance and upgrades can be avoided by switching users from the node to be worked on to other nodes in the application network so that their service continues while their original node is being upgraded.

GoldenGate Director

The GoldenGate Director provides configuration and management of multiple host platforms from a single console. It runs in its own server and provides a GUI management interface either directly or over the Web. It can integrate other 3rd party monitoring facilities via built-in or custom adapters.

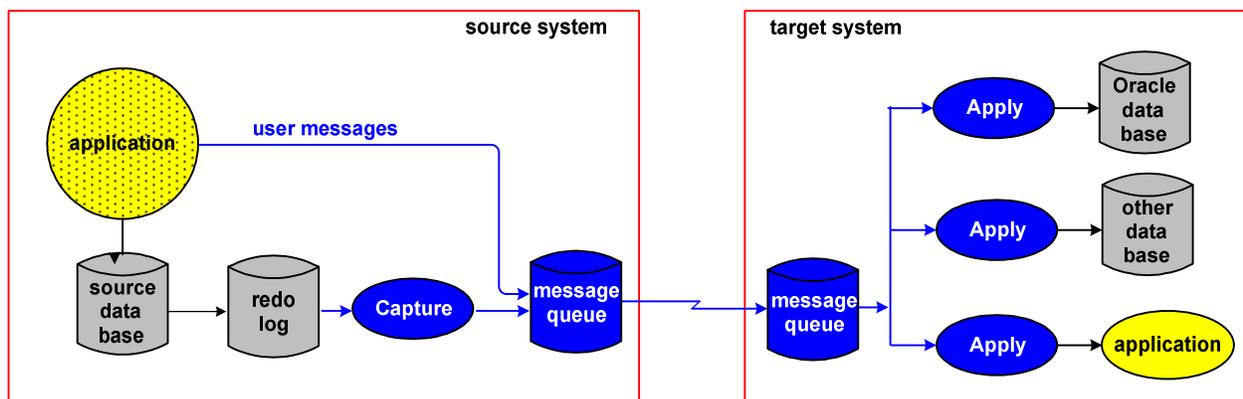
GoldenGate Veridata

GoldenGate Veridata is a stand-alone product that identifies and reports on discrepancies between two distinct operational databases. It does this by comparing two databases online with no interruption to the applications. Verification is accomplished by comparing row signatures. Veridata provides reports of discrepancies for manual review and correction.

Oracle Streams

Oracle Streams enables the sharing of messages via a message stream.⁶ The stream can propagate information to other databases or applications. Stream rules route specified information to specified destinations.

Streams accomplishes data replication by propagating changes from a source database to a destination database. Streams can be configured for bidirectional data replication, and it provides data collision detection and resolution.



Oracle Streams

⁶ Introduction to Streams, Oracle white paper.
http://download.oracle.com/docs/cd/B10501_01/server.920/a96571/strmover.htm

Architecture

Changes made to tables, schemas, or an entire database are recorded in the Oracle redo log for the database. A Capture process fetches changes from the redo log and formats each change into a message, which is placed in a persistent queue. Users and applications can also enqueue messages. Streams propagates messages from one queue to another. The destination queues can be in the same database or in different databases.

An Apply process at the destination dequeues messages from a queue. The Apply process can apply messages directly to destination databases, or it can pass messages to custom programs for processing.

The Capture Process

Changes made to database objects in an Oracle database are logged in the redo log to guarantee recoverability in the event of user error or media failure. A Capture process formats these changes into messages and enqueues them. There are two types of messages. One type contains information about a change to a row in a table resulting from a DML (data manipulation language) operation. The other contains information about a DDL (data definition language) change to a database object.

The Apply Process

An Apply process dequeues messages from a destination queue and either applies each message directly to a destination database or passes the message to a user-defined procedure. User defined procedures include message handlers, DML handlers, DDL handlers, precommit handlers, and error handlers.

Typically, an Apply process applies messages to the local database where it is running; but in a heterogeneous database environment, it can be configured to apply messages to a remote non-Oracle database.

User Messaging

A user or application can also enqueue and dequeue messages. Oracle Streams enables user applications to enqueue messages into a queue, propagate messages to subscribing queues, notify user applications that messages are ready for consumption, and dequeue messages at the destination. Streams supports all the standard features of message queuing systems, including multiconsumer queues, publish and subscribe, content-based routing, Internet propagation, transformations, and gateways to other messaging subsystems.

Data Replication

For data replication purposes, Streams can capture DML and DDL changes made to a source database and replicate those changes to one or more destination databases. A Streams Capture process captures changes made to source database objects and formats them into messages that can be propagated to destination queues, from where they are then applied by Streams Apply processes.

Streams can replicate changes bidirectionally between databases. The tables for which data is shared do not need to be identical copies at all databases. Both the structure and the contents of these tables can differ at different databases.

Configuration

Data Protection

Data protection is achieved by creating a remote copy of a production database. In the event of human error or a catastrophe, the copy can be used to resume processing. This capability can also be accomplished with Oracle Data Guard, which uses some of the same infrastructure as Streams.

Rules

Streams Capture processes and Apply processes perform actions based on rules. Rules specify which events are captured, propagated, applied, and dequeued. The ability to capture events and propagate them to relevant consumers based on rules means that Streams can be used for event notification. Messages representing events can be staged in a queue and dequeued explicitly by a messaging client or an application. Actions can then be taken based on these events, such as email notification or sending a message to a cell phone or pager.

Data Warehouse Loading

Data warehouse loading is a special case of data replication. Some of the most critical tasks in creating and maintaining a data warehouse include refreshing existing data and adding new data from the operational databases. Streams components can capture changes made to a production system and send those changes to a staging database or directly to a data warehouse or operational data store.

Eliminating Planned Downtime

The features of Oracle Streams can be used to achieve little or no database downtime during database upgrade and maintenance operations. Maintenance operations include migrating a database to a different platform, migrating a database to a different character set, modifying database schema objects to support upgrades to user-created applications, and applying an Oracle software patch.

Automatic Conflict Detection and Resolution

An Apply process detects conflicts automatically when applying messages in a bidirectional replication environment. A conflict results when the same row in the source database and the destination database is changed at approximately the same time.

Streams offers a variety of prebuilt conflict handlers to resolve conflicts. In addition, users can build their own conflict resolution handlers. If a conflict is not resolved or if a handler procedure raises an error, then all messages in the transaction that raised the error are saved in the error queue for later analysis and possible reexecution.

Oracle to Non-Oracle Data Sharing

To share DML changes from an Oracle source database with a non-Oracle destination database, the Oracle database functions as a proxy and carries out some of the steps that would normally be done at the destination database. The messages intended for the non-Oracle destination database are dequeued in the Oracle database itself, and an Apply process at the Oracle database applies the messages to the non-Oracle database.

To capture and propagate changes from a non-Oracle database to an Oracle database, a custom application is required at the non-Oracle source database. The application fetches changes made to the non-Oracle database by reading from transaction logs, by using triggers, or via some other method. The application must assemble and order the transactions and must convert each change into a message.

Next, the application must enqueue the messages into a queue in an Oracle database, where they can be processed by a Streams Apply process.

Summary

Oracle is making a major shift in its data replication strategy, moving from Oracle Streams to its newly acquired GoldenGate replication engine. Industry conjecture is that the heterogeneous capabilities of GoldenGate that are lacking in Streams give Oracle a powerful way to integrate its database into environments that currently do not use Oracle. GoldenGate will provide the mechanism for Oracle to become the database of record in these environments, accepting data from other databases and distributing its centralized data to those databases.