Transparent Role Transitions With Oracle Data Guard and Oracle GoldenGate

ORACLE WHITE PAPER | FEBRUARY 2015

Table of Contents

Introduction

To achieve the highest levels of availability resulting in zero or near-zero downtime for both unplanned outages and all planned maintenance activities, customers frequently use the combination of Oracle Active Data Guard (Oracle ADG) and Oracle GoldenGate. Oracle Data Guard provides zero data loss failover, comprehensive data protection and very fast failover for database failures, cluster failures and data corruptions. Using Oracle Active Data Guard offers the additional benefit of being able to open the standby database in read-only mode, allowing queries to return results that are identical to what would be returned from the primary database. The other significant Oracle Active Data Guard HA benefit is auto block repair for physical block corruptions. Oracle GoldenGate provides advanced replication features, including bi-directional replication used to facilitate zero downtime planned maintenance, migrations, and for database and application upgrades. Oracle Active Data Guard and Oracle GoldenGate are essential components of the Gold and Platinum MAA reference architectures as described in Oracle MAA Architecture – The Foundation for Database as a Service and the *Oracle Database 12c High Availability Overview* guide.

This Maximum Availability Architecture (MAA) best practices paper describes the configuration best practices to enable Oracle GoldenGate replication using a database that is protected by a Data Guard standby to transparently and seamlessly work following an Oracle Data Guard role transition, no matter which Data Guard protection mode is configured (Maximum Performance, Availability or Protection).

The key prerequisites are:

- An Oracle GoldenGate database that is a source or target database for Oracle GoldenGate replication that is protected with Oracle Data Guard or Oracle Active Data Guard.
  The Oracle GoldenGate database can have a subset of or all of the following Oracle GoldenGate components running against it: Extract, Data Pump and Replicat. This is only applicable for Oracle Database 11*g* and later releases using Oracle Data Guard or Oracle Active Data Guard with physical standby databases.

- Oracle Grid Infrastructure 11*g* Release 2 (11.2.0.4) or later.
  Oracle Grid Infrastructure provides the necessary components needed to manage high availability for any business-critical applications. Using Oracle Clusterware (a component of

Oracle Grid Infrastructure) network, database and Oracle GoldenGate resources can be managed to provide availability in the event of a failure.

- Oracle Grid Infrastructure Agent version 6.1 or later.
  The Oracle Grid Infrastructure Agent leverages the Oracle Grid Infrastructure components to provide integration between Oracle GoldenGate and its dependent resources, such as the database, network and file system. The agent also integrates Oracle GoldenGate with Oracle Data Guard so that Oracle GoldenGate is restarted on the new primary database following a role transition.

- Oracle Database 11*g* Release 2 (11.2.0.4) or later.
  Use Oracle Database 11*g* Release 2 (11.2.0.4) or later to take advantage of Oracle GoldenGate integrated Extract and integrated Replicat features. Refer to My Oracle Support Note 1557031.1 for a full list of recommended Oracle database patches when using Oracle GoldenGate.

- Oracle GoldenGate version 12.1.2.0 or later.
  Use Oracle GoldenGate Extract in integrated capture mode deployed locally. Oracle also recommends using Oracle GoldenGate integrated Replicat, available with Oracle Database 11*g* Release 2 (11.2.0.4) or later.

- Oracle DBFS to protect and replicate critical GoldenGate files.
  The Oracle Database File System (DBFS) is the only MAA validated and recommended file system because it allows the storage of the important Oracle GoldenGate files, such as the checkpoint and trail files, to be located inside the same database that is protected with Oracle Data Guard, ensuring consistency between the Oracle GoldenGate files and the database. A DBFS disk group is also pre-configured with Oracle Exadata Database Machines.

Oracle highly recommends using Oracle Clusterware and the Oracle Grid Infrastructure Agent to simplify and automate the integration of Oracle GoldenGate and Oracle Data Guard. If you cannot use Oracle Clusterware and the Oracle Grid Infrastructure Agent, refer to the instructions provided in the Appendix of this paper.

For all generic MAA and Oracle GoldenGate MAA best practices, refer to the following URL: http://www.oracle.com/goto/maa.

MAA Architecture Examples With Oracle GoldenGate and Oracle Active Data Guard

For some, an active-active HA database architecture is required to take advantage of the unplanned and planned maintenance advantages with Oracle GoldenGate. However, Oracle GoldenGate alone cannot guarantee zero data loss in the case of database, cluster or data corruption failures. To protect against these types of failures, the MAA architectures illustrated in the following figures are suggested.

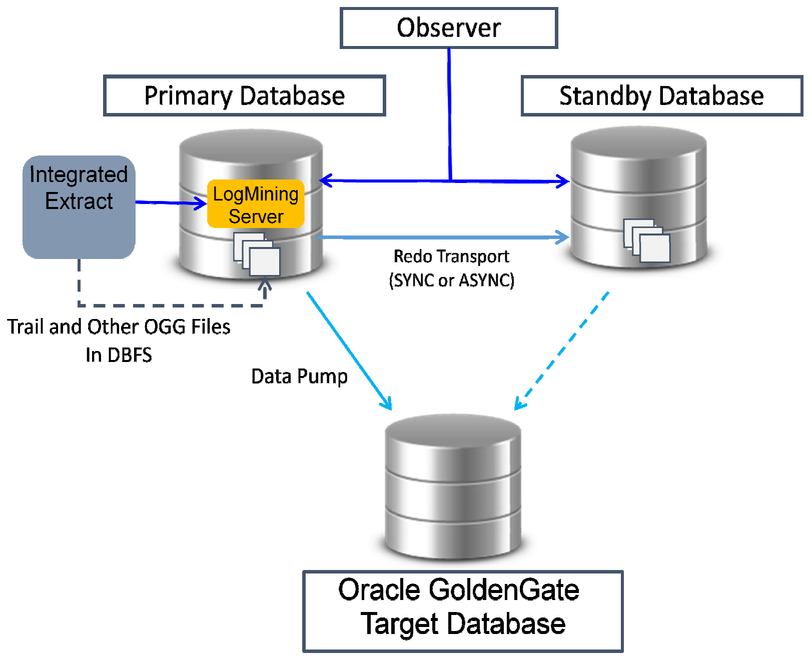
Observer
Primary Database
Standby Database
Integrated Extract
LogMining Server
Redo Transport (SYNC or ASYNC)
Trail and Other OGG Files In DBFS
Data Pump
Oracle GoldenGate Target Database

Figure 1: Example Architecture 1

In Figure 1, when executing a failover or switchover to the physical standby database, the Oracle GoldenGate Extract and Target databases work seamlessly after the Oracle Data Guard role transition. The Oracle Active Data Guard standby database is the primary failover target in case there are database or cluster failures. Oracle Active Data Guard is read-only and active, but all application transactions that contain DMLs or DDLs must be redirected to the primary database. The Oracle GoldenGate target database is used to mitigate the downtime for planned maintenance activities, such as an application upgrade.

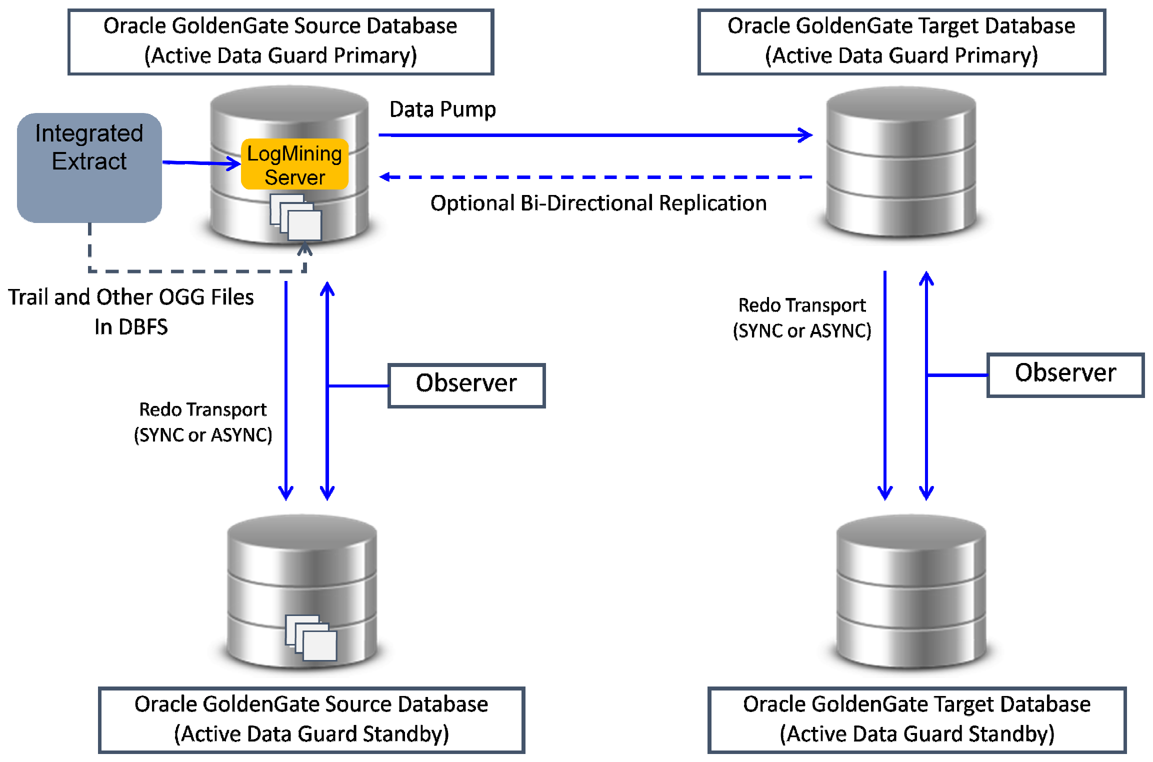
Oracle GoldenGate Source Database
(Active Data Guard Primary)
Oracle GoldenGate Target Database
(Active Data Guard Primary)
Integrated Extract
LogMining Server
Data Pump
Optional Bi-Directional Replication
Trail and Other OGG Files In DBFS
Redo Transport (SYNC or ASYNC)
Observer
Redo Transport (SYNC or ASYNC)
Observer
Oracle GoldenGate Source Database
(Active Data Guard Standby)
Oracle GoldenGate Target Database
(Active Data Guard Standby)

Figure 2: Example Architecture 2

In Figure 2, the two primary databases are replicating data between each other using Oracle GoldenGate. The data replication can be uni-directional or bi-directional. When data loss cannot be tolerated, Oracle Active Data Guard fast-start failover configured in maximum availability or maximum protection mode (`SYNC` transport) is configured for each Oracle GoldenGate database. When fast-start failover is enabled, the Oracle Data Guard broker and its observer determines if a failover is necessary and initiates the failover to the specified standby database automatically, with no need for DBA intervention. After Oracle Data Guard failover or switchover, Oracle GoldenGate replication will continue to work seamlessly.

Figure 2 depicts Oracle GoldenGate being active on the primary database. The post-failover diagram in Figure 3 shows that Oracle GoldenGate has been re-enabled on the new primary database at the standby site.

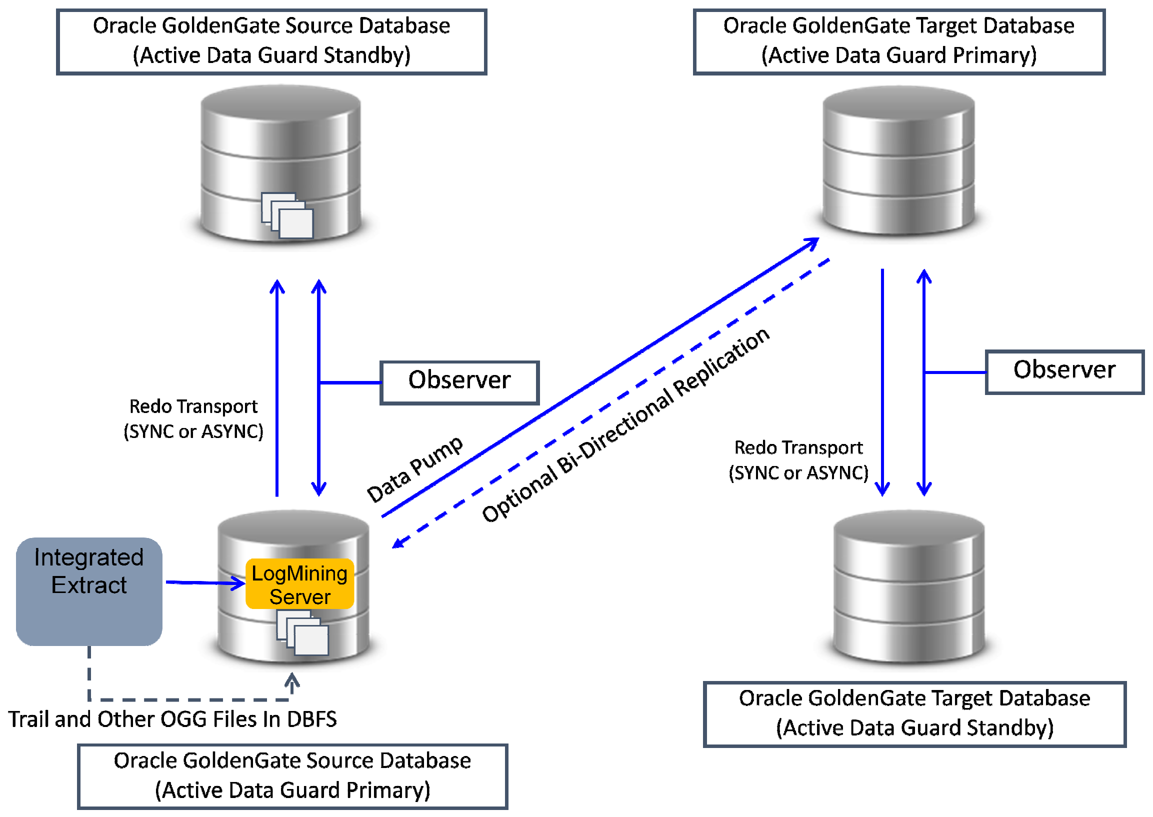
Oracle GoldenGate Source Database (Active Data Guard Standby)
Oracle GoldenGate Target Database (Active Data Guard Primary)
Observer
Redo Transport (SYNC or ASYNC)
Data Pump
Optional Bi-Directional Replication
Observer
Redo Transport (SYNC or ASYNC)
Integrated Extract
LogMining Server
Trail and Other OGG Files In DBFS
Oracle GoldenGate Source Database (Active Data Guard Primary)
Oracle GoldenGate Target Database (Active Data Guard Standby)

Figure 3: Post-Failover of the Oracle GoldenGate Source Database

Configuring DBFS for Oracle GoldenGate Shared Files

During an Oracle Data Guard role transition, it is critical that Oracle GoldenGate maintains its checkpoint and process information. MAA recommends storing these Oracle GoldenGate files on DBFS mount points. The DBFS tablespaces reside in the same Oracle GoldenGate database that is protected by Oracle Data Guard so the database and Oracle GoldenGate files remain synchronized after an Oracle Data Guard role transition.

Follow the DBFS configuration best practices included in the *Oracle GoldenGate with Oracle Real Application Clusters Configuration* white paper located at the following URL:

http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf

High-level DBFS configuration requirements are:

- Create the two DBFS tablespaces in the primary database that is protected by Oracle Data Guard; one for the Oracle GoldenGate checkpoint files and the second for all other Oracle GoldenGate files.
- The DBFS LOB segment created for the checkpoint files file system is set to `CACHE LOGGING` and the other one is set to `NOCACHE LOGGING`.
- Both file systems are owned by the same database user, so a single `mount` command mounts both file systems.

As recommended in the MAA white paper, *Oracle GoldenGate with Oracle Real Application Clusters Configuration* (http://www.oracle.com/technetwork/database/features/availability/maa-goldengate-rac-2007111.pdf), all of the Oracle GoldenGate files that need to be accessible from another host should be placed in DBFS. This includes the trail files (`dirdat`), checkpoint files (`dirchk`), parameter files (`dirprm`), bounded recovery files (`BR`) , credential store files (`dircrd`) and temporary files (`dirtmp`).

NOTE: When using DBFS and Oracle GoldenGate with Oracle Data Guard, the process files directory (`dirpcs`) must remain on a local disk and not on DBFS. Placing the process files on DBFS can cause problems during Oracle Data Guard failovers and switchovers.

Oracle GoldenGate Integrated Capture

Integrated Capture mode Extract is required in order to continue extracting through an Oracle Data Guard role transition. Classic Capture does not permit automatic mining through a role transition and, therefore, is not recommended. Integrated Capture provides the following benefits:

- Supports asymmetric Oracle Data Guard configurations where the primary and standby configurations may have a different number of Oracle RAC nodes and instances. Oracle recommends using a symmetric primary and standby database configuration.
- Supports extracting through database `RESETLOGS` operations, such as those typical with an Oracle Data Guard failover operation.
- Extracts the redo changes from the primary database once the redo has been written to both the primary online redo logs and the standby redo logs on the standby database. This ensures that replication stays synchronized after a zero data loss Oracle Data Guard failover.

In addition, integrated Capture supports more Oracle database features than classic Capture, such as compressed data (Basic, OLTP and Exadata Hybrid Columnar Compression) and improved Oracle RAC support. Integrated Extract integrates with RMAN fast recovery area retention policies to ensure that archive logs are not removed until Extract no longer needs them.

For detailed configuration best practices for Oracle GoldenGate integrated Extract, refer to the following URL:

http://www.oracle.com/technetwork/database/availability/maa-gg-performance-1969630.pdf

Recommendation highlights:

- Enable database force logging to ensure that all changes are found in the redo and are recoverable.
- Enable database minimal supplemental logging and create additional supplemental log groups through the use of the Oracle GoldenGate command `ADD SCHEMATRANDATA`.
- Configure the streams pool with the initialization parameter `STREAMS_POOL_SIZE`.
- Install the `UTL_SPADV` package and the Oracle GoldenGate integrated features healthcheck for integrated Extract and Replicat configuration and performance analysis.

In preparation for the Extract process to run on the primary or standby database hosts, Oracle recommends setting an explicit connection string in the Extract parameter file using the Oracle GoldenGate credential store or an encrypted password. Oracle does not recommend using plain text password in the Extract parameter file.

Example GoldenGate credential store Extract parameter:

```
USERIDALIAS ggconn
```

The `ggconn` credential alias name is added to the credential store using the following:

```
GGSCI> ALTER CREDENTIALSTORE ADD USER ggadmin@ggtns, password ggadmin, alias ggconn
```

Example Encrypted password Extract parameter:

```
USERID ggadmin@ggtns, password AACAAAAAAAAAAAMAVJSBMDNBWHDJCAEGDJUIAEHBYIOACAJJ
BLOWFISH ENCRYPTKEY DEFAULT
```

Make sure the Transparent Network Substrate (TNS) alias specified in the `tnsnames.ora` parameter file connects to the source database running on that particular node. The TNS alias, `ggtns` in this example, is different on each node, but shares the same alias name. The TNS entry in the `tnsnames.ora` parameter file must include a service name for the role-based service created later in this paper. The role-based service is only started on the host of the current primary database that is running, and the Oracle GoldenGate Extract process will only run on the database if the role-based service is running.

Oracle Data Guard Failover With Data Loss Considerations

For Data Guard configurations where there is always the potential for data loss (Data Guard Maximum Performance Mode using `ASYNC` Redo Transport), the following parameter must be added to the integrated Extract process parameter file:

```
TRANLOGOPTIONS HANDLEDLFAILOVER
```

This parameter prevents Extract from extracting redo data and writing to the trail file data that has not yet been applied to the Oracle Data Guard standby database. This is crucial in preventing Oracle GoldenGate from replicating data to a target database that does not exist in the standby database. If this parameter is not specified, after a data loss failover, it is possible to have data in the target database that is not present in the source database leading to logical data inconsistencies.

After a role transition, if the old primary (or the new standby) database is not available, Extract will not be able to mine any redo data until the Oracle GoldenGate target standby database is available. The Extract process may abend with the following error:

```
2015-01-23 18:28:00  ERROR   OGG-02077  Extract encountered a read error in the
asynchronous reader thread and is abending: Query to retrieve applied SCN of the
target standby database failed.
```

If the Manager process restart parameters have been specified correctly as described in the “Oracle GoldenGate Manager” section later in this paper, the Extract process will attempt to restart.

If the old primary database will be unavailable for an extended period of time or completely gone, then in order for Oracle GoldenGate replication to continue, you must remove the `HANDLEDLFAILOVER` parameter from the Extract parameter file. Extract no longer waits until redo has been applied to the standby database before extracting the data. During the time it takes for the standby database to come back online and apply all the redo from the primary database there will be data divergence between it and the Oracle GoldenGate target database. This will be resolved once the standby database is up to date and the `HANDLEDLFAILOVER` parameter is added back into the integrated Extract process parameter file.

Oracle also recommends adding this parameter to Data Guard configurations using Maximum Availability mode with `SYNC` redo transport. Under usual operation, Maximum Availability results in a zero data loss failover when an outage occurs and, in such cases, this parameter is not necessary. There is always the possibility, however, of data loss when multiple outages occur. One such example is a use-case where there is a network outage between the primary and standby database, but the primary continues to process new transactions, and then a second outage causes the primary to fail some time later before the first problem is resolved. Any data generated by new transactions since the first outage occurred is lost. Setting this parameter is required to address this use-case.

Oracle recommends configuring Fast-Start Failover (FSFO) so that the broker can automatically fail over to a previously chosen standby database in the event of loss of the primary database. Without FSFO, failing over to the standby database is a manual process.

If Oracle Data Guard Fast-Start Failover is disabled, you must specify an additional integrated Extract parameter. For example:

```
TRANLOGOPTIONS _FAILOVERTARGETDESTID n
```

This parameter identifies which standby database the Oracle GoldenGate Extract process must remain behind, with regards to not extracting redo data that has not yet been applied to the Oracle Data Guard standby database. To determine the correct value for `_FAILOVERTARGETDESTID`, the `archive_log_dest` database initialization parameter is used. Replace `n` with the correct archive log destination identifier.

For example:

```
SQL> show parameters log_archive_dest

NAME                                 TYPE        VALUE

------------------------------------ ----------- ------------------------------

log_archive_dest_1                   string      location=USE_DB_RECOVERY_FILE_

                                                 DEST, valid_for=(ALL_LOGFILES,

                                                  ALL_ROLES)

log_archive_dest_2                   string      service="ggs2d", ASYNC NOAFFIR

                                                 M delay=0 optional compression

                                                 =disable max_failure=0 max_con

                                                 nections=1 reopen=300 db_uniqu

                                                 e_name="GGS2D" net_timeout=30,

                                                  valid_for=(online_logfile,all

                                                 _roles)
```

In this example, the Extract parameter would be set to the following:

```
TRANLOGOPTIONS _FAILOVERTARGETDESTID 2
```

Oracle GoldenGate Data Pump

The Oracle GoldenGate Data Pump process reads the source trail files and, after any filtering or transformations, transmits them to the target host where the collection server process writes the data to the local trail files. This should not be confused with Oracle's Data Pump (`impdp` or `expdp`) utility. By default, the way in which trail files are copied from the source to the target host is by transferring small packets of data before the trail file write on the target host is issued. Confirmation that the data has successfully been written to disk (using `fsync` on Linux) does not occur very frequently which, in the event of a role transition, opens up the possibility of the trail file getting unsynchronized with a checkpoint from one or more Replicat processes. To reduce the chances of this happening, the Data Pump must be configured with the following parameters:

```
RMTHOST <hostname>, MGRPORT <port#>, PARAMS -f -B1048576, TCPBUFSIZE 1048576,
TCPFLUSHBYTES 1048576
```

The combination of the `TCPBUFSIZE` and `TCPFLUSHBYTES` parameters increases the transfer buffer between the source and target to 1MB, with the collector process on the target issuing a write for every 1MB (`-B` write buffer size parameter) received, followed directly with a write confirmation (`-f` flush and sync).

When the target database of an Oracle GoldenGate environment, where Replicat processes run, is protected with an Oracle Data Guard configuration, there is an important consideration that must be given to the Data Pump running on the source environment. The host that the Data Pump transfers the source trail files to is determined by the `RMTHOST` parameter. Oracle recommends using a virtual IP (VIP) address for the remote host name which can be moved between the primary and standby hosts. When a VIP cannot be defined for a primary and standby host in an Oracle Data Guard environment, there is a need for additional configuration to ensure that the source Data Pump always sends trail files to the current primary host. These additional configuration steps are detailed in My Oracle Support Note 1950121.1.

If your Data Guard configuration uses maxmium performance or maximum availability protection modes, apply the GoldenGate patch for bug 19870326. In the event of a data loss Data Guard failover, the Oracle GoldenGate target trail files can be smaller than expected at the site of the new primary database. This patch (19870326) allows the Oracle GoldenGate Data Pump to restart after an Oracle GoldenGate target database Data Guard data loss failover.

In preparation for the Data Pump process to run on the primary or standby database hosts, Oracle recommends setting an explicit connection string in the Data Pump parameter file using the Oracle GoldenGate credential store, as described in the "Oracle GoldenGate Integrated Capture" section earlier in this paper.

Oracle GoldenGate Replicat

Oracle recommends using integrated Replicat, introduced in Oracle GoldenGate Release 12.1 and Oracle Database 11*g* Release 2 (11.2.0.4). Integrated Replicat leverages the apply process functionality that is available inside the database. The integrated Replicat checkpoint information, required for process recovery after a failure, is automatically maintained in the database. Therefore, configuring a checkpoint table is not required.

If using non-integrated Replicat processes, the processes must be configured to use a checkpoint table so that the checkpoints are recorded in the database instead of relying on the checkpoint file. Instructions for creating the checkpoint table are provided in the *Installing and Configuring Oracle GoldenGate for Oracle Database* white paper located at the following URL:

http://docs.oracle.com/goldengate/1212/gg-winux/GIORA/config_apply.htm#GIORA931

For higher performance configuration for Replicat, refer to the MAA white paper *Oracle GoldenGate Performance Best Practices* located at the following URL:

http://www.oracle.com/technetwork/database/availability/maa-gg-performance-1969630.pdf

After an Oracle Data Guard failover, if the non-integrated Replicat fails to start due to an error with the checkpoint being larger than the trail file, the following error message is reported:

```
2014-05-16 11:44:04  ERROR   OGG-01705  Input checkpoint position 388369723 for input
trail file '/u01/goldengate/dirdat_os/bb000028' is greater than the size of the file
(387583957).  Please consult Oracle Knowledge Management Doc ID 1138409.1. for
instructions.
```

If using a checkpoint table, which is recommended with non-integrated Replicat, the process can be restarted with the following command:

```
GGSCI> START REPLICAT <replicat name> FILTERDUPTRANSACTIONS
```

Refer to My Oracle Support Note 1536741.1 for more detailed process restart instructions. Integrated Replicat automatically handles recovery after an Oracle Data Guard role transition, so this error will not be reported.

In preparation for the Replicat process to run on the primary or standby database hosts, Oracle recommends setting an explicit connection string in the Data Pump parameter file using the Oracle GoldenGate credential store, as described in the "Oracle GoldenGate Integrated Capture" section earlier in this paper.

Oracle Data Guard Failover With Data Loss Considerations

After an Oracle Data Guard failover that resulted in some data loss, even though the database has lost data, the tables being updated by the Oracle GoldenGate Replicat processes will not lose any data. This is because any missing trail file data stored in DBFS in the target database will be resent by the source Data Pump process. Data inconsistencies may exist between objects not replicated by Oracle GoldenGate and those being replicated.

Oracle GoldenGate Manager

The Manager process must be configured so that it automatically restarts the Oracle GoldenGate processes. Set the following Manager parameters in the `mgr.prm` parameter file:

```
AUTOSTART ER *
AUTORESTART ER *, RETRIES 20, WAITSECONDS 15, RESETMINUTES 60
```

`AUTOSTART` automatically starts the Extract, Data Pump and Replicat processes after the Manager process starts. `AUTORESTART` tries restarting any Oracle GoldenGate processes after they fail. To reduce the time it takes for the Oracle GoldenGate processes to restart after a role transition, Oracle recommends setting the maximum number of retries to 20 and the wait time between each retry to 15 seconds.

Oracle Cluster Resource (CRS) Configuration

For complete automation of the Oracle GoldenGate process shutdown and startup with Oracle Data Guard, create the following CRS resources.

Create the Oracle Clusterware Role-Based Service for Oracle GoldenGate

Using a role-based service allows Oracle GoldenGate and other resources needed by Oracle GoldenGate, like DBFS, to be started after an Oracle Data Guard role transition. The service is only started on the host of the current primary database that is running.

As the Oracle user, create an Oracle GoldenGate role-based service using the following command (Oracle Database 12c example):

```
% srvctl add service -db GGS1 -service oggserv -role PRIMARY -preferred GGS21
-available GGS22
```

Notes:

- The database service, `GGS1` in this example, should already be created for the database. Refer to the *Oracle Database Administrator's Guide* found at the following URL:

  http://docs.oracle.com/database/121/ADMIN/restart.htm#ADMIN5009

- `-role PRIMARY` indicates this service can only be started when the database is an Oracle Data Guard primary database.
- `-preferred` is only required if the database and role-based service is to run on a subset of the available cluster nodes. Naming a database instance with `-preferred` ensures that the service has a preference to start on this database instance in the Oracle RAC cluster.
- `-available` lists the other Oracle RAC database instance on which the role-based service is able to run.

If using Oracle Database 11g release 2 (11.2.0.4) create an Oracle GoldenGate role-based service using the following command:

```
% srvctl add service -d GGS1 -s oggserv -l PRIMARY -r GGS21 -a GGS22
```

Start and check the status of the new role-based service using the following commands (Oracle Database 12c example):

```
% srvctl start service -db GGS1 -service oggserv

% srvctl status service -db GGS1 -service oggserv
```

If using Oracle Database 11g release 2 (11.2.0.4) use the following commands to start and check the new role-based service:

```
% srvctl start service -d GGS1 -s oggserv

% srvctl status service -d GGS1 -s oggserv
```

Create the `oggserv` role-based service on both of the Oracle Data Guard primary and standby hosts.

The TNS alias defined in the `tnsnames.ora` parameter file must connect to the database using the role-based service name. For example:

```
ggconn=
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = gghost1)(PORT = 1521))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = oggserv)
   )
 )
```

For further information on creating a database service, refer to the *Oracle Database Administrator's Guide* found at the following URL:

http://docs.oracle.com/database/121/ADMIN/restart.htm#ADMIN5009

Create the DBFS Resource

Creating a CRS resource for DBFS provides automatic mounting and unmounting of the DBFS mount points required by Oracle GoldenGate by the Oracle Grid Infrastructure Agent (detailed in a later step).

Use the `mount-dbfs.sh` action script and follow the instructions in My Oracle Support Note 1054431.1. Store the action script in the same location on all Oracle RAC cluster nodes and all Oracle Data Guard standby hosts.

NOTE: Modify the `mount-dbfs.sh` action script to force unmounting of DBFS which prevents Oracle Data Guard switchovers from hanging or failing. Change all `fusermount -u` commands to `fusermount -uz`.

Test the `mount-dbfs.sh` action script for mounting DBFS. For example:

```
% ./mount-dbfs.sh start
% ./mount-dbfs.sh status

Checking status now
Check – ONLINE

% df -k
```

If you cannot see the DBFS mount point, refer to My Oracle Support Note 1054431.1 for troubleshooting.

Test unmounting of DBFS. For example:

```
% ./mount-dbfs.sh stop
% ./mount-dbfs.sh status

Checking status now
Check – OFFLINE

% df -k
```

Create the DBFS CRS resource with the following command:

```
#!/bin/bash

ACTION_SCRIPT=/u01/oracle/scripts/mount-dbfs.sh

RESNAME=dbfs_mount

DEPNAME=ora.ggs1.oggserv.svc

ORACLE_HOME=/u01/app/12.1.0.2/grid

PATH=$ORACLE_HOME/bin:$PATH

export PATH ORACLE_HOME

crsctl add resource $RESNAME \

  -type cluster_resource \

  -attr "ACTION_SCRIPT=$ACTION_SCRIPT, \

         CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \

         START_DEPENDENCIES='hard($DEPNAME)pullup($DEPNAME)',\

         STOP_DEPENDENCIES='hard($DEPNAME)',\

         SCRIPT_TIMEOUT=300"
```

Change the following variables:

- `ACTION_SCRIPT` – specifies the location and name of the `mount-dbfs.sh` action script.
- `DEPNAME` – specifies the full name of the `oggserv` role-based service created in a step described earlier in this paper. To determine the full name, issue the `crsctl stat res|grep oggserv`.
- `ORACLE_HOME` – specifies the Gird Infrastructure home directory.

Create Oracle GoldenGate Application Virtual IP (VIP) Address If Using Oracle RAC

An application virtual IP (VIP) address is a cluster resource that Oracle Clusterware manages. The VIP is assigned to a single cluster node and is migrated to another node in the cluster in the event of a node failure. This allows Oracle GoldenGate Data Pump to continue transferring data to the newly assigned target Oracle RAC node.

When Oracle GoldenGate is registered with the Oracle Grid Infrastructure Agent in the next section of this white paper, an application VIP is automatically created. If an application VIP already exists with the same IP address, it must first be dropped.

Oracle Grid Infrastructure Agent Configuration

The Oracle Grid Infrastructure Agent leverages the Oracle Grid Infrastructure components to provide integration between Oracle GoldenGate and its dependent resources, such as the database, network and file system. The agent also integrates Oracle GoldenGate with Oracle Data Guard so that Oracle GoldenGate processes can be restarted on the new primary database following a role transition.

Oracle Grid Infrastructure version 12*c* already includes a version of the agent called a bundled agent. This is not the latest version available. Oracle recommends downloading the latest version (v6.1) of the standalone agent from the following URL:

http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/index.html

Before installing the standalone agent, you must remove the older version of the agent if it is installed outside of the Oracle Grid Infrastructure home directory.

NOTE: The downloadable standalone agents cannot be applied to the Oracle Grid Infrastructure Bundled Agent home (for example, `$GRID_HOME/xag`).

Be sure to check the version support matrix found in the Oracle Grid Infrastructure Agents documentation. The current version of the agent, 6.1, is certified with Oracle Grid Infrastructure version 11.2.0.3 and later versions, including 12.1.

Make sure the location of the newly installed agent `bin` directory is listed in the `PATH` environment variable before the Oracle Grid Infrastructure home directory. For example:

```
% export PATH=/u01/oracle/bundledagent/xag/bin:$PATH
```

As the `root` user, create the Oracle GoldenGate agent configuration with the following command:

```
agctl add goldengate GG_PRMY --gg_home /home/oracle/goldengate/ \

--oracle_home /u01/app/oracle/product/12.1.0.2/dbhome_1 \

--db_services ora.ggs1.oggserv.svc --monitor_extracts ext_1a,dpump_1a \

--monitor_replicats rep_1a,rep_2a \

--environment_vars
'TNS_ADMIN=/u01/app/oracle/product/12.1.0.2/dbhome_1/network/admin' \

--filesystems dbfs_mount \

--nodes RACnode1,RACnode2 --dataguard_autostart yes --user oracle \

--group oinstall --network 1 --ip 192.168.0.54
```

Notes:

- The resource name, `GG_PRMY`, can be named something more meaningful, if desired.
- `gg_home` and `oracle_home` must be set to your correct environment settings.
- Specify the previously created `oggserv` role-based service with the `db_services` parameter.
- Specify the DBFS CRS resource name with the `filesystems` parameter created in an earlier step described in this white paper.
- List the Oracle RAC nodes where Oracle GoldenGate can run with the `nodes` parameter.

- Specifying `dataguard_autostart` automatically starts and stops Oracle GoldenGate after an Oracle Data Guard role transition. You must also specify `user` and `group` of the Oracle GoldenGate operating system user.
- Specify the network number for the application VIP with the `network` parameter. You can determine the network number with the following command:

  ```
  % crsctl stat res -p |grep -ie .network -ie subnet |grep -ie name -ie subnet
  ```

  Consider the following sample output:

  ```
  NAME=ora.net1.network
  USR_ORA_SUBNET=10.1.41.0
  ```

  `net1` in `NAME=ora.net1.network` indicates this is network `1`, and the second line indicates the subnet on which the VIP will be created.
- Specify the IP address of the application VIP with the `ip` parameter. The application VIP is created with the naming format `xag.<resource_name>-vip.vip`. For example: `xag.GG_PRMY-vip.vip.`

Once Oracle GoldenGate is registered with the agent, `agctl` should be used for stopping and starting the Oracle GoldenGate processes.

To start Oracle GoldenGate and the dependent resources (application VIP, `oggserv` role-based service and DBFS), issue the following command as the Oracle user:

```
% agctl start goldengate GG_PRMY
```

To relocate Oracle GoldenGate and the dependent resources (application VIP, `oggserv` role-based service and DBFS) to another Oracle RAC node in the cluster, issue the following:

```
% agctl relocate goldengate GG_PRMY --node <node_name>
```

To check the status of Oracle GoldenGate, issue the following:

```
% agctl status goldengate GG_PRMY
Goldengate  instance 'GG_PRMY' is running on RACnode1
```

To stop all Oracle GoldenGate processes, including Manager, issue the following:

```
% agctl stop goldengate GG_PRMY
```

Stopping Oracle GoldenGate does not automatically stop the dependent resources. To shutdown all of the dependent resources after Oracle GoldenGate is shutdown, issue the following:

```
% srvctl stop service -d <DB name> -s oggserv
```

```
# Run the following two commands as the Oracle Grid Infrastructure user:

% crsctl stop resource xag.GG_PRMY-vip.vip

% crsctl stop resource dbfs_mount
```

Now that Oracle GoldenGate is configured with the Oracle Grid Infrastructure Agent, Oracle GoldenGate restarts automatically on the primary database after a role transition, or on a surviving Oracle RAC node after a node failure.

Conclusion

The combination and integration of Oracle GoldenGate and Oracle Data Guard enables customers to achieve a Platinum MAA service-level configuration that achieves zero or near zero downtime for all planned and unplanned outages. An integral part of this solution is that Oracle GoldenGate can work transparently after a zero data loss Oracle Data Guard role transition. With the configuration and operational steps described in this paper, Oracle GoldenGate can be configured to work seamlessly with Oracle Data Guard after any zero data loss or data loss role transition. By using DBFS as the file system for important Oracle GoldenGate files, such as trail files, checkpoint files, and bounded recovery files, Oracle GoldenGate Extract and Replicat processes continue to stay synchronized with the database after a role transition.

Appendix: Oracle GoldenGate Configuration Without the Oracle Grid Infrastructure Agent

When the Oracle Grid Infrastructure Agent cannot be used, seamless integration between Oracle GoldenGate and Oracle Data Guard is handled through the use of a database role transition trigger.

The following sections replace the sub-sections in the "Oracle Cluster Resource (CRS) Configuration" section found earlier in this paper. The Oracle GoldenGate processes (Manager, Extract, Data Pump, and Replicat) configuration remains the same whether or not you use the Oracle Grid Infrastructure Agent.

Database Configuration for Automatic DBFS Mounting and Unmounting

There are a number of triggers that are required to enable DBFS to be automatically mounted and unmounted when the database is opened or shut down.

NOTE: All of the shell scripts that are created in this section must also exist in the same location on the standby and primary hosts. If this is not done, the role transition trigger fails when the scripts cannot be found. Execute the following steps:

1. Unmount DBFS on database shut down.

   Before the database is shut down, the DBFS mount should be unmounted so that the stale mount point is not left on the system when the database is not open.

   a. Create the following shell script for unmounting the DBFS mount point (owned by the database user, typically `oracle`, with permissions set to 6571):

```
#!/bin/sh
# unmount_dbfs.sh - stops all GoldenGate processes and unmounts DBFS
# First stop all GoldenGate processes:

/home/oracle/scripts/stop_all_GG.sh

# Now unmount DBFS mount point:
fusermount -uz /mnt/dbfs

exit 0
```

```
#!/bin/sh
# stop_all_GG.sh - stops all the GoldenGate processes, including Manager

export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
export ORACLE_SID=GGS1
OGG_HOME=/home/oracle/goldengate
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib

cd $OGG_HOME
./ggsci << !EOT
```

```
stop er *!
kill er *
stop mgr!
exit
!EOT

exit 0
```

b. Set the permissions to 6571 on these scripts. Execute the scripts manually to make sure they work:

```
% df -k|grep dbfs

% ./umount_dbfs.sh

% df -k|grep dbfs
```

c. Create the following database job and database shutdown trigger (as `SYSDBA`):

```
SQL> exec
dbms_scheduler.create_job(job_name=>'dbfs_unmount',job_type=>'EXECUTABLE',
job_action=>'/home/oracle/scripts/unmount_dbfs.sh',enabled=>FALSE);
```

d. Make sure DBFS is mounted, run the job, and then check to see that DBFS is unmounted:

```
SQL> exec dbms_scheduler.run_job(job_name=>'dbfs_unmount');
```

e. Create the database shutdown trigger:

```
SQL> create or replace trigger dbfs_unmount_trigger
before shutdown on database
declare
  role varchar(30) ;
begin
  select database_role into role from v$database ;
  if role = 'PRIMARY' then
    dbms_scheduler.RUN_JOB('dbfs_unmount');
  end if;
end;
/
```

f. Test the trigger by mounting DBFS and then shutting down the database.

NOTE: The database shutdown trigger is executed with a `SHUTDOWN NORMAL` or `SHUTDOWN IMMEDIATE` command only. The trigger will not be executed with a `SHUTDOWN ABORT` command.

2. Mount DBFS on database startup.

DBFS should also be automatically mounted after the database is opened in preparation for the Oracle GoldenGate processes to start.

NOTE: Make sure that you have already configured `/etc/fstab` to allow DBFS mounting using the `MOUNT` command as described in a step earlier in this white paper. To mount DBFS on database startup, execute the following steps:

a. Create the following shell script for mounting the DBFS mount point (owned by the database user, typically `oracle`, with permissions set to 6571):

```
#!/bin/sh
# mount_dbfs.sh - mounts DBFS

export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
export ORACLE_SID=GGS1
OGG_HOME=/home/oracle/goldengate
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib

# Force DBFS unmounting incase mount point remains from previous
# failure

fusermount -uz /mnt/dbfs
mount /mnt/dbfs

# If desired, add code in here to restart GoldenGate processes

exit 0
```

b. Create the following database job and database startup trigger (as `SYSDBA`):

```
SQL> exec
dbms_scheduler.create_job(job_name=>'dbfs_mount',job_type=>'EXECUTABLE',
job_action=>'/home/oracle/scripts/mount_dbfs.sh',enabled=>FALSE);
```

c. Make sure DBFS is mounted, run the job and then check to see that DBFS is unmounted:

```
SQL> exec dbms_scheduler.run_job(job_name=>'dbfs_mount');
```

d. Create the database startup trigger:

```
SQL> create or replace trigger dbfs_mount_trigger
after startup on database
declare
  role varchar(30) ;
begin
  select database_role into role from v$database ;
  if role = 'PRIMARY' then
```

```
    dbms_scheduler.RUN_JOB('dbfs_mount');
end if;
end;
/
```

e. Test the trigger by dismounting DBFS and then starting the database.

3. Mount DBFS and start Oracle GoldenGate processes when a role transition occurs from standby to primary databases.

During a role transition when the standby database becomes a primary database, the database is not shut down and restarted and the shutdown and startup triggers do not get executed. Therefore, a separate role transition trigger is required. This only gets executed when the standby database becomes a primary. To mount DBFS and start the Oracle GoldenGate processes when a role transition occurs, execute the following steps:

a. Create the following shell script for mounting the DBFS mount point and starting the Oracle GoldenGate Manager and Oracle GoldenGate processes (owned by the database user, typically `oracle`, with permissions set to 6571):

```
#!/bin/sh
# failover_actions.sh - Mounts DBFS and starts GoldenGate on a
#                      database role transition.

export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
export ORACLE_SID=GGS1
OGG_HOME=/home/oracle/goldengate
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib

# Now unmount (call twice to be sure DBFS is unmounted) and
# mount DBFS:

fusermount -uz /mnt/dbfs
fusermount -uz /mnt/dbfs
mount /mnt/dbfs

# Start GoldenGate:
cd ${GGS_HOME}
./ggsci <<EOFF
stop er *!
stop mgr!
pause 2
start mgr
exit
EOFF

exit 0
```

Be sure to test the script to ensure DBFS is mounted and the GoldenGate processes are started.

b. Create the following database job and database role transition trigger (as SYSDBA):

```
exec
dbms_scheduler.create_job(job_name=>'ogg_failover',job_type=>'EXECUTABLE',
job_action=>'/home/oracle/scripts/failover_actions.sh',enabled=>FALSE);
```

c. Test the script executes on the primary database using the scheduled job:

```
SQL> exec dbms_scheduler.run_job(job_name=>'ogg_failover');
```

d. Create the database role transition trigger:

```
create or replace trigger ogg_failover
after db_role_change on database
declare
  role varchar2(30);
begin
  select database_role into role from v$database ;
  if role = 'PRIMARY' then
    dbms_scheduler.RUN_JOB('ogg_failover');
  end if;
end;
/
```

The only way to test this trigger is through a role transition.

Oracle GoldenGate Data Pump Configuration and Role Transitions

When the target database of an Oracle GoldenGate environment where Replicat processes run is protected with an Oracle Data Guard configuration, there is an important consideration that must be given to the Data Pump running on the source environment. The host that the Data Pump transfers the source trail files to is determined by the RMTHOST parameter. Oracle recommends using a virtual IP (VIP) address for the remote host name which can be moved between the primary and standby hosts. When a VIP cannot be defined for a primary and standby host in an Oracle Data Guard environment, additional configuration is needed to ensure that the source Data Pump always sends trail files to the current primary host. These additional configuration steps are detailed in My Oracle Support Note 1950121.1.

The role transition trigger provide in My Oracle Support Note 1950121.1 should be modified to also include the DBFS mounting command and the Oracle GoldenGate process start commands before the source Data Pump configuration is changed.

For example:

```
create or replace trigger ogg_failover
after db_role_change on database
declare
  role varchar2(30);
  hostname varchar2(64);
begin
  select database_role into role from v$database;
  select host_name into hostname from v$instance;
  if role = 'PRIMARY' then
    dbms_scheduler.RUN_JOB('ogg_failover');
    if hostname = '<primary host name/VIP>'
    then
      change_rmthost@gg_source('PRIMARY');
    elsif hostname = '<standby host name/VIP>'
    then
      change_rmthost@gg_source('STANDBY');
    end if;
  end if;
end;
/
```

ORACLE®

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Transparent Role Transitions with Oracle Data Guard and Oracle GoldenGate

January 2015

Author: Stephan Haisley
Contributors: Lawrence To

Hardware and Software, Engineered to Work Together