Oracle Maximum
Availability Architecture

Oracle Active Data Guard Far Sync
Zero Data Loss at Any Distance

Oracle Maximum Availability Architecture Best Practices

ORACLE WHITE PAPER | AUGUST 2014

Table of Contents

Introduction

Oracle Active Data Guard is the most comprehensive solution available to eliminate single points of failure for mission critical Oracle Databases. It prevents data loss and downtime in the simplest and most economical manner by maintaining a synchronized physical replica(s) of a production database at one or more remote locations. If the production database is unavailable for any reason, applications can quickly, and in some configurations transparently, failover to the synchronized replica to restore service.

Active Data Guard also eliminates the high cost of idle redundancy by allowing reporting applications, ad-hoc queries, and data extracts to be offloaded to read-only copies of the production database. Active Data Guard's deep integration with Oracle Database and complete focus on real-time data protection and availability avoids compromises found in alternative data protection solutions.

Data Guard is the only Oracle Database replication solution that can provide both zero data loss protection and near-immediate restoration of service should a production database become unrecoverable for any reason. This is accomplished using the combination of Data Guard synchronous redo transport and a replication-aware apply process at the standby database.

The impact that any synchronous replication method can have on database performance, however, often makes it impractical to implement zero data loss protection when large distances separate the primary and replica database(s). Rather than impact database performance, many enterprises will compromise on data protection by implementing asynchronous replication and accept that an unrecoverable outage will result in varying degrees of data loss.

Active Data Guard Far Sync, a new capability with Oracle Database 12c, eliminates compromise by extending zero data loss protection to a replica database located at any distance from the primary database. Active Data Guard Far Sync accomplishes this with minimal expense or complexity compared to other extended distance data protection and availability solutions. This paper describes Oracle Maximum Availability Architecture (Oracle MAA) best practices and operational insight useful for successfully implementing Active Data Guard Far Sync to achieve zero data loss protection over any distance.

Data Guard Zero Data Loss Protection

Zero data loss protection requires synchronous communication between a production database (primary) and the replica used for data protection and availability (a standby database at a remote destination). The most popular zero data loss configuration using Data Guard or Active Data Guard uses Maximum Availability protection mode with synchronous redo transport (SYNC).

SYNC transport combined with a deep understanding of Oracle transaction semantics by Data Guard apply services provides a guarantee of zero data loss during unplanned outages of a primary database. As users commit transactions at a primary database, Oracle generates redo records and writes them to a local online log file. Data Guard transport services simultaneously transmits the same redo directly from the primary database log buffer (memory allocated within system global area) to the standby database(s) where it is written to a standby redo log file. SYNC transport requires that the primary database wait for both the local and remote log file writes to complete before commit success is signaled to the application. As the distance between primary and standby increases, however, the total round-trip time required to acknowledge the remote log file write can reach a point where it has too great of an impact on database performance to make it practical to support zero data loss protection.

Prior to Oracle Database 12c, the only solution for achieving zero data loss protection across a wide area network was to deploy a Data Guard configuration with two standby databases. The first standby is located within a distance from the primary where the round-trip network latency makes it practical for synchronous communication. The second standby is deployed at the remote location and utilizes asynchronous communication (ASYNC). Zero data loss failover to the standby at the remote location is a two-step process: a zero data loss failover to the local standby followed by a zero data loss switchover (planned event) to the remote standby where production will run. This solution has a number of advantages, however, the added expense and complexity of a second standby database can be difficult to justify if its only purpose is for zero data loss failover to the remote location.

Active Data Guard Far Sync

New in Oracle Database 12c, Active Data Guard Far Sync provides the ability to perform a zero data loss failover to a remote standby database without requiring a second standby database or complex operation. Far Sync enables this by deploying a Far Sync instance (a lightweight Oracle instance that has only a control file, spfile, password file and standby log files, there are no database files or online redo logs) at a distance that is within an acceptable range of the primary for SYNC transport. A Far Sync instance receives redo from the primary via SYNC transport and immediately forwards the redo to up to 29 remote standby databases via ASYNC transport as described in Figure 1. A Far Sync instance can also forward redo to the new Oracle Database Backup, Logging, and Recovery Appliance[1].

The presence of a Far Sync instance in a Data Guard configuration is transparent to its operation during switchover or failover, the administrator uses the same commands used for any Data Guard configuration. Far Sync requires nothing new to learn or any additional procedures in order to perform a zero data loss failover across a wide area network (WAN).

A Far Sync instance offloads the primary of any overhead for any of the following tasks:

- Resolving gaps in archived logs received by the remote standby database(s) (e.g. following network or standby database outages).
- Redo transport overhead for configurations having multiple standby databases. The primary ships once to the Far Sync instance, Far Sync takes care of shipping to multiple destinations.

[1] http://www.oracle.com/us/corporate/features/database-backup-logging-recovery-appliance/index.html

» A Far Sync instance can also perform off-host network compression, conserving WAN bandwidth without impacting primary database performance.

The lightweight nature and transparent operation of the Far Sync instance is a substantial improvement over the previous multi-standby solution to WAN zero data loss protection; there are no user data files, no media recovery and no Oracle Database license required for the Far Sync instance. The Far Sync instance only needs sufficient disk space for standby redo log files and to retain archived redo logs to resolve any gaps that might occur. The Far Sync instance requires a very small SGA footprint (much less than production) and consumes less than one CPU (except when it also performs redo transport compression).

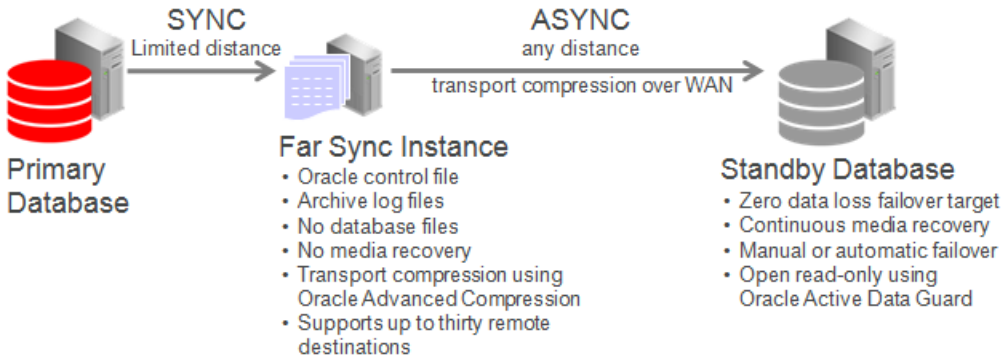

Figure 1: Active Data Guard Far Sync Architecture

Far Sync provides increased flexibility in the location of a disaster recovery site for those who wish to implement zero data loss protection. Even users who have already deployed Data Guard synchronous transport can benefit from configuring a Far Sync instance closer to the primary than their current standby to reduce the performance impact on the production database.

Far Sync also benefits users who currently use asynchronous transport. Upgrading to zero data loss protection using Far Sync eliminates the uncertainty and administrative burden that accompanies the need to reconcile data loss after a failover has occurred in an asynchronous configuration. Far Sync can increase availability by eliminating the tendency to postpone failover in the hope of resolving an outage without losing data. Far Sync's guarantee of zero data loss encourages immediate failover to quickly resume service while the problems that caused the outage are resolved. The primary in an ASYNC configuration also enjoys the same benefits of offloading the overhead of gap resolution, servicing multiple standby databases, and redo transport compression, described above.

In almost every case, Far Sync is ideal for achieving the highest level of data protection with the least performance impact while also lowering network bandwidth consumption.

Note that an Active Data Guard license includes the right to use Far Sync. The primary database and any standby database that receives redo from a Far Sync instance must be licensed for Active Data Guard. The Active Data Guard license also includes the right to deploy a Far Sync instance on single-instance (non-RAC) database without requiring a separate Oracle Database license. If Far Sync is deployed on Oracle RAC, however, a separate Oracle RAC license is required.

Far Sync Deployment Architectures - Examples

There are several common use-cases for Far Sync.

Case 1: Simple Configuration with a Far Sync Instance

This is the most basic example in which a Primary database uses a single Far Sync instance to extend zero data loss failover to a remote standby database (Figure 2). Ideally the Far Sync instance is deployed in a separate location than the primary database in order to isolate it from site failure, but within a metro area distance (network RTT of 5ms or less – subject to performance testing). Even if no separate location is available there is still a benefit to deploying a Far Sync instance within the same data center to enable fast, zero data loss failover for all unrecoverable outages short of full site failure.

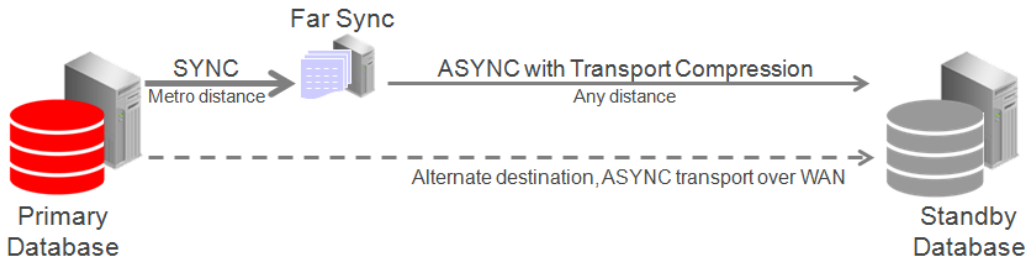

Figure 2: Simple Far Sync with ASYNC Alternate Destination

In this simple use-case there is only a single Far Sync instance configured with the remote standby defined as an alternate destination on the primary database. This enables Data Guard to automatically use asynchronous transport directly to the remote destination to maintain near-zero data loss protection should there be an outage of the Far Sync instance. Data Guard will automatically return the configuration to zero data loss protection once the Far Sync instance has been repaired and connection re-established.

Case 2: Far Sync HA

This builds upon the previous example by adding a second Far Sync instance as an alternate destination for the first (Figure 3). Data Guard will automatically switch from one Far Sync instance to the next with minimal disruption to zero data loss protection. HA for the Far Sync instance in this context refers to the ability to maintain zero data loss protection. An outage of a Far Sync instance never has any impact on the availability of the primary or standby databases.

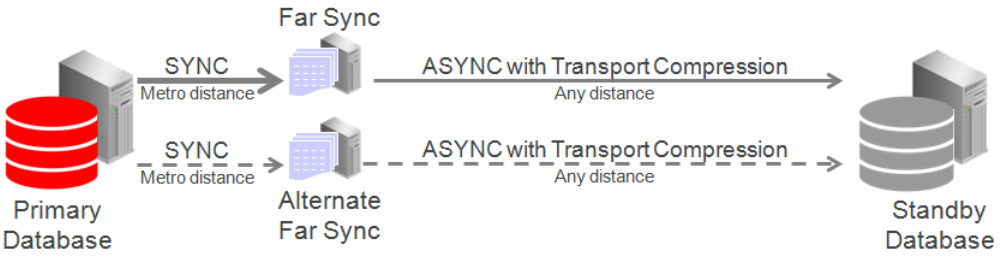

Figure 3: HA Configuration for Zero Data Loss Protection

Not pictured, but an effective alternative for HA is to deploy Far Sync on Oracle RAC (this requires an Oracle RAC license). Oracle RAC can eliminate or minimize the period of time when the configuration falls out of a zero data loss protection level during a Far Sync outage. More details are provided in a later section of this paper.

Case 3: Zero Data Loss Protection Following Role Transitions

This builds upon Cases 1 and 2 by using an additional Far Sync instance that is within a metro area distance of the remote standby database (Figure 4). The remote Far Sync instance is idle while the standby is in standby role. It becomes active when the standby transitions to the primary database role enabling zero data loss failover to the new standby (old primary). The Far Sync that is local to the original primary becomes inactive while it is in standby role.

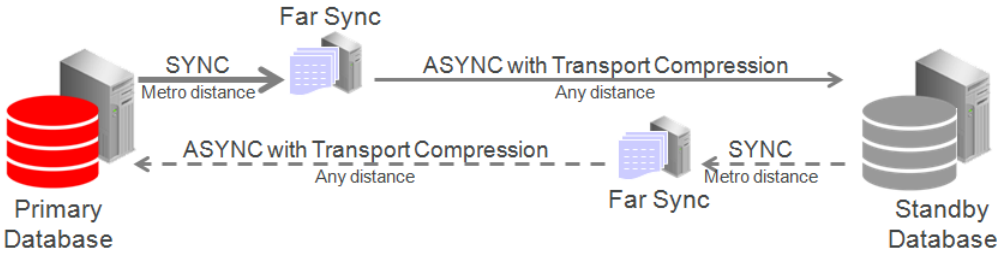

Figure 4: Zero Data Loss Protection Following Role Transition

Case 4: Reader Farm

Far Sync can support up to 30 remote destinations. This makes it a very useful tool for supporting a reader farm – an Active Data Guard configuration having multiple active standby databases to easily scale read performance. In this example the reader farm is configured in a remote location from the primary database (Figure 5). The primary ships once over the WAN to the Far Sync instance located in the remote destination and Far Sync distributes redo locally to all active standby databases in the reader farm.

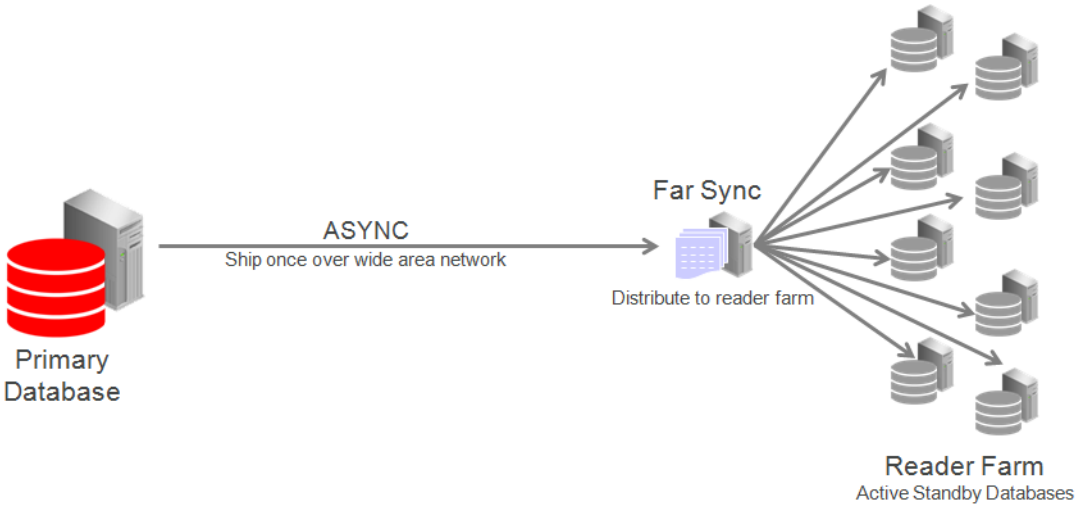

Figure 5: Reader Farm – Ship Once and Distribute to Many

The above example shows only one approach for configuring a reader farm. For example, the primary database can also have a standby for disaster recovery. Following a switchover or failover, the new primary will automatically resume redo shipment to the Far Sync instance. The DR standby may also be placed in the same location as the Reader Farm. The primary can ship once to the DR standby, and the standby can cascade redo to the Far Sync instance for distribution to the reader farm.

Case 5: Cloud Deployment - Far Sync Hub

Far Sync is a very lightweight process. A single physical server can support multiple Far Sync instances, each providing zero data loss failover to a remote destination. An example is shown in Figure 6 where primary databases ship to a single physical machine operating as a Far Sync Hub – multiple Far Sync instances on a single physical machine. Primaries and Far Sync Hub are on premises with standby databases deployed remotely in the cloud. Note that all systems in such a configuration (primary and standby database hosts and Far Sync host) must meet the usual requirements for compatibility in a Data Guard configuration described in My Oracle Support Note 413484.1.

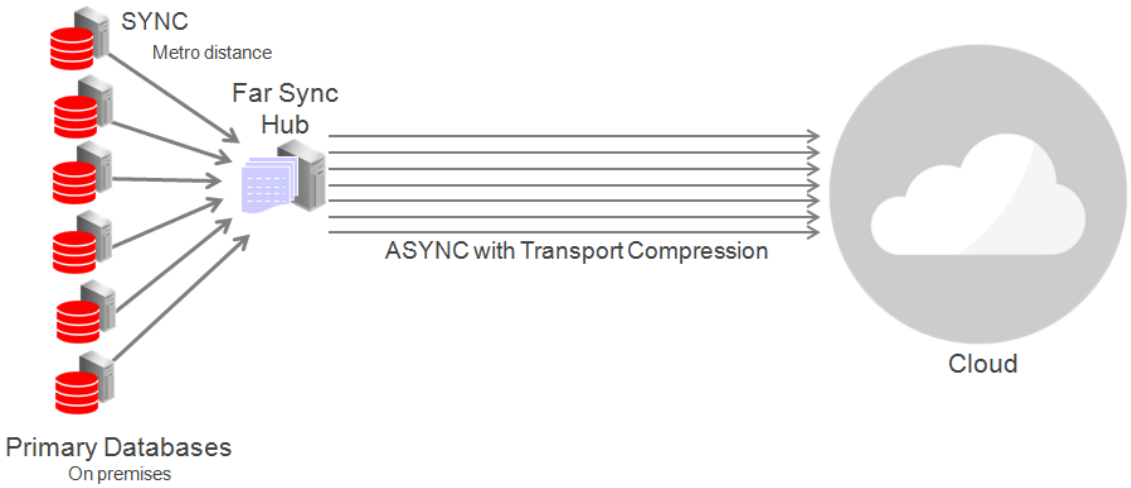

Figure 6: Cloud Deployment

Far Sync Configuration Best Practices

A Far Sync instance behaves just like any other Data Guard/Active Data Guard archive destination configured for SYNC transport. This means that the laws of physics regarding the speed of light that affect round-trip network latency have not been altered in any way; primary database performance will still be impacted by the network round-trip time between the primary database and the Far Sync instance. Far Sync, however, enables flexible options for achieving zero data loss over long distance by placing a Far Sync instance at a metro distance from the primary database and allowing the Far Sync instance to handle all communication with a remote standby that can be located hundreds or thousands of miles away. This enables zero data loss failover for all cases except for a disaster that impacts a wide geographic area causing an outage of both the primary and the Far Sync instance at the same time.

There are few configuration best practices necessary. Most are the same as those that would apply to any SYNC redo transport destination. They are:

- The network between the primary database and far sync instance must:
  - Have round trip latency low enough so that the impact to response time and throughput of the primary database does not exceed business requirements. The degree of impact is very application specific and will require testing to validate. In general, experience shows that there is a higher likelihood of success if the round-trip latency is less than 5ms, though there are successful deployments at higher latencies.
  - Provide enough bandwidth between the primary database and far sync instance to accommodate peak redo volumes in addition to any other traffic sharing the network. Redo transport compression can be used to reduce network bandwidth requirements.
  - Ideally there are redundant network links that will also tolerate network component failure.
- Standard Data Guard network best practices, such as the appropriate setting of TCP send and receive buffer sizes equal to three times the bandwidth delay product. For more information see MAA practices for Data Guard network transport.[2]
- A Far Sync instance's standby redo logs (SRLs) should be placed on storage with sufficient IOPS (writes per second) capacity to exceed the I/O of the LGWR process on the primary database during peak activity in addition to any IOPS from other activities. This is an important consideration. For example:
  - If the Far Sync instance has lesser performing disks than the primary it may not be able to forward redo to remote destinations as fast as it is received and an archive log gap can form.
  - In the case of redo gap resolution scenarios due to planned maintenance on the standby or network outages for example, there will be additional IO requests for gap resolution on top of redo from current transactions.
  - Lesser performing disks at the Far Sync instance will delay acknowledgement to the primary database, increasing the total round-trip time between primary and standby and impacting application response time. This impact can be eliminated by using Fast Sync between the primary and the Far Sync instance (see details on Fast Sync later in this paper).
- The Far Sync instance's standby redo logs (SRLs) should have the same number of redo log groups as on the primary +1 for each thread as per standard MAA Best Practices.
- The SRLs of an alternate Far Sync should be manually cleared prior to use in order to achieve the fastest return to SYNC transport and zero data loss protection when an alternate Far Sync is activated.
- Performance testing has shown that a small Far Sync instance SGA does not impact performance of the Far Sync instance nor the Primary database. The MAA recommendation is to configure the minimum SGA required for Far Sync to function.
  - Use instance caging[3] in order to achieve the smallest possible SGA. Reducing the CPU_COUNT during testing had no effect on the performance of the Far sync instance.
  - MAA tests determined that a 300MB SGA 300 with CPU_COUNT=1 on Linux was sufficient for Far Sync.(CPU_COUNT=1 SGA_TARGET=300M)
- Configure the RMAN archivelog deletion policy at the Far Sync instance to 'SHIPPED TO STANDBY' or 'APPLIED ON STANDBY' to automatically manage disk space on the Far Sync instance. Backing up the archive logs at the Far Sync instance is not necessary provided there is a proper backup plan for the primary or standby.
- Configure Far Sync instances for both the primary and standby databases to allow for zero data loss protection to be maintained following role transitions. The Far Sync instance deployed within a metro distance of the standby database is idle until the standby becomes primary.
  - Note that in a Data Guard Broker configuration a switchover (planned role transition) cannot occur while in Maximum Availability unless the protection mode can be enforced from the target standby site. If the standby

[2] http://docs.oracle.com/cd/E11882_01/server.112/e10803/config_dg.htm#HABPT5284
[3] http://docs.oracle.com/database/121/ADMIN/dbrm.htm#ADMIN13333

does not have its own Far Sync instance it will have to be configured to ship ASYNC to the original primary after roles are reversed. This will prevent a switchover from occurring unless the protection mode for the primary database is first dropped from Maximum Availability to Maximum Performance.

- Fast Sync, a new capability with Data Guard 12c, improved primary database performance compared to SYNC between 4% and 12% depending on the network latency and the I/O speed of the Far Sync instance hardware. Please refer to the later section of this paper that describes Fast Sync.
- Placing the Far Sync instance on a virtual machine produced no difference in performance over physical hardware configurations provided CPU, I/O and network requirements are met.

Configuring Far Sync – Examples and Outage Scenarios

An outage of a Far Sync instance running in Maximum Availability mode has no impact on the availability of the primary database other than a transitory brown-out while the primary database receives error notification. The primary resumes processing database transactions after notification is received. In most cases notification is immediate though there are certain fault conditions that suspend fault notification until the threshold for net_timeout expires (a user configurable Data Guard parameter with a default of 30 seconds). This is standard operation for any Data Guard configuration that uses Maximum Availability mode. Data Guard's self-healing mechanisms will automatically reconnect and resynchronize a standby database once the problem that caused it to disconnect is resolved. These same automatic mechanisms also apply to Far Sync.

HA in the context of Far Sync addresses the ability to eliminate or minimize the potential for data loss should there be a double failure scenario, e.g. a primary database outage immediately following a Far Sync outage or vice versa. HA in this context can be achieved in multiple ways. Each approach has its own trade-offs and considerations and is described in the sections that follow. Combinations of the options listed are also feasible.

Note: Creation of a Far Sync instance is documented in Section 5.1, Data Guard Concepts and Administration.[4]

HA using the Terminal Standby as an Alternate Destination

The simplest approach to maintaining data protection during a Far Sync outage is to create an alternate LOG_ARCHIVE_DEST_N pointing directly to the terminal standby (the ultimate failover target), similar to the architecture depicted in Figure 2. Asynchronous transport (ASYNC) to the remote destination is the most likely choice in order to avoid the performance impact caused by WAN network latency. ASYNC can achieve near-zero data loss protection (as little as sub-seconds of exposure) but because it never waits for standby acknowledgement it is unable to provide a zero data loss guarantee. During a Far Sync outage, redo transport will automatically failover to using the alternate destination. Once the Far Sync instance has been repaired and resumes operation, transport will automatically switch back to the Far Sync instance and zero data loss protection is restored.

Note that during a switchover to the terminal standby (a planned role transition) the protection mode of the configuration must be reduced to Maximum Performance so that the mode is enforceable on the target of the role transition. Changing protection modes and transport methods is a dynamic operation with zero downtime.

[4] http://docs.oracle.com/database/121/SBYDB/create_fs.htm#SBYDB5416

The characteristics of this approach include:

- No additional Far Sync hardware or instances to manage.
- Loss of zero data loss coverage during a far sync instance outage. Data protection level drops to UNSYNCHRONIZED with ASYNC transport until the Far Sync instance is able to resume operation, synchronous communication is re-established and the standby become fully synchronized.

Relevant configuration parameters for this example include:

- Primary (primary):

  log_archive_dest_2= service="farsync", SYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=5 db_unique_name="farsync" net_timeout=8, alternate=LOG_ARCHIVE_DEST_3 valid_for=(online_logfile,all_roles)

  log_archive_dest_3 =service="standby", ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=5 db_unique_name="standby" alternate=LOG_ARCHIVE_DEST_2 valid_for=(online_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_dest_state_3=ALTERNATE

  log_archive_config= dg_config=(primary, farsync, standby)

  fal_server= standby

- Primary Far Sync "A" (farsync)

  log_archive_dest_2=" service="standby", ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=0 max_connections=1 reopen=5 db_unique_name="standby" net_timeout=8, valid_for=(standby_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_config= dg_config=(primary, farsync, standby)

  fal_server=primary

- Standby (standby):

  log_archive_dest_2= service="primary" ASYNC reopen=5 db_unique_name="primary" valid_for=(online_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_config= dg_config=(primary,farsync, standby)

  fal_server=farsync, primary

*The above parameters are relevant for this configuration when set by SQL*Plus and are provided to make all the details of the configuration clear to the reader, however Oracle recommends the simpler approach of configuring Far Sync using the Data Guard Broker. Steps for configuring Far Sync with Data Guard Broker are listed in Appendix C*

For Oracle RDBMS Version 12.1 set the parameter "_redo_transport_stall_time"=60 in all instances within the configuration for best return to synchronization after a node outage with an ALTERNATE Far Sync configuration.

HA using an Alternate Far Sync Instance

A more robust approach to maintaining data protection in the event of a Far Sync outage is to deploy a second Far Sync instance as an alternative destination (Figure 3). Unlike the previous example, the alternate Far Sync instance

prevents the data protection level from being degraded to Maximum Performance (ASYNC) while the first Far Sync instance is repaired.

Figure 7 shows what occurs during an outage when the active Far Sync instance in this example has failed but the server on which it was running is still operating:

- There is a brief application brownout of 3 to 4 seconds while the primary database connects to the alternate Far Sync instance.
  - In this failure state an error is immediately returned to the primary database, avoiding net_timeout. The brownout can be calculated based upon parameter settings ((max_failure-1)*reopen) + 4s.
- The maximum potential data loss should there be a second outage that impacts the primary database before the configuration is resynchronized is a function of the amount of redo generated while the primary reconnects to the alternate Far Sync instance and the configuration is resynchronized.

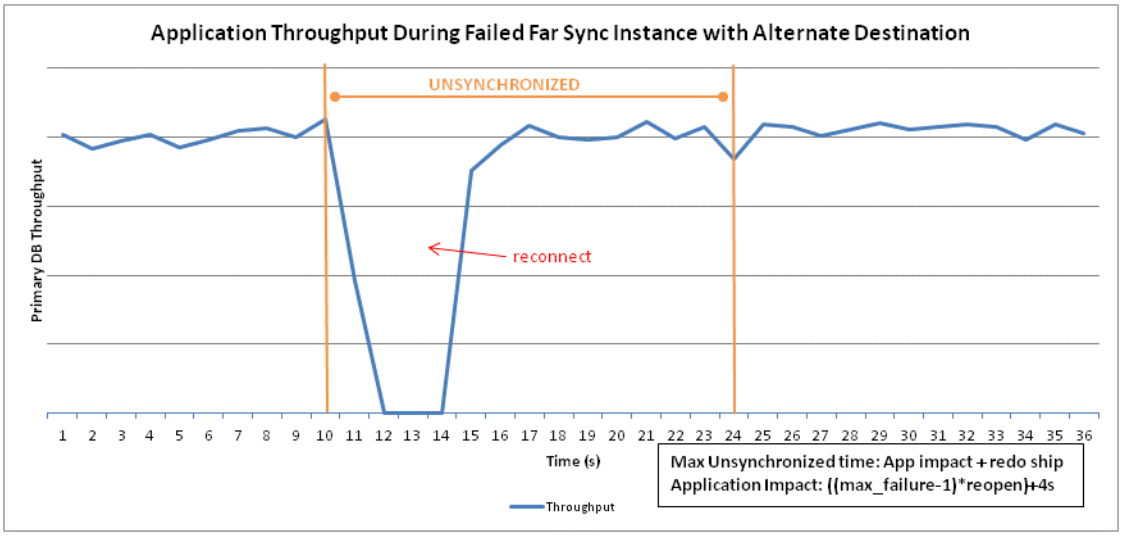

Figure 7 – A Far Sync instance outage using a second Far Sync instance as an alternate destination

Behavior is slightly different in a failure state where the server on which the primary Far Sync instance is running goes down hard. The application brownout will be extended by an additional net_timeout seconds and the potential exposure to data loss should a second outage impact the primary is increased due to additional processing that is required before resynchronization is complete.

This use-case also assumes a symmetric configuration of Far Sync instances within metro area distance of the remote standby database to enable zero data loss failover in the other direction should there be a planned role transition, as described in a previous section of this paper, 'Case 3 , Zero Data Loss Protection Following Role Transitions'.

The characteristics of this approach include:

- Reduced data loss exposure should there be a second outage that impacts the primary database before a Far Sync instance is repaired and returned to service.
- A symmetrical configuration that is able to maintain zero data loss protection following a role transition (planned switchover) that promotes the standby database to the primary role (the original primary becomes a zero data loss failover target). Note that starting in 12.1 a SQL*Plus configuration with an alternate destination will

automatically fall back to the initial log archive destination when it becomes available. Within a Data Guard broker configuration the FALLBACK keyword in the 'redoroutes' property is used to enable this behavior[5].

Relevant configuration parameters for this example include:

- Primary (primary):

  log_archive_dest_2= service="prifarsyncA" SYNC NOAFFIRM max_failure=1 reopen=5 db_unique_name="prifarsyncA" net_timeout=8, alternate=LOG_ARCHIVE_DEST_3 valid_for=(online_logfile,all_roles)

  log_archive_dest_3 =service="prifarsyncB" SYNC NOAFFIRM max_failure=1 reopen=5 db_unique_name="prifarsyncB" net_timeout=8 alternate=LOG_ARCHIVE_DEST_2 valid_for=(online_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_dest_state_3=ALTERNATE

  log_archive_config= 'dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB)

  fal_server=sbfarsyncA, sbfarsyncB, standby

- Primary Far Sync "A" (prifarsyncA)

  log_archive_dest_2=" service="standby" ASYNC reopen=5 db_unique_name="standby" valid_for=(standby_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_config= 'dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB)

  fal_server=primary

- Far Sync "B" (prifarsyncB)

  log_archive_dest_2=" service="standby", ASYNC reopen=5 db_unique_name="standby" valid_for=(standby_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_config= 'dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB)

  fal_server=primary

- Standby (standby):

  log_archive_dest_2= service="sbfarsyncA", SYNC NOAFFIRM max_failure=1 reopen=5 db_unique_name="sbfarsyncA" net_timeout=8, alternate=LOG_ARCHIVE_DEST_3 valid_for=(online_logfile,all_roles)

  log_archive_dest_3 =service="sbfarsyncB", SYNC NOAFFIRM max_failure=1 reopen=5 db_unique_name="sbfarsyncB" net_timeout=8, alternate=LOG_ARCHIVE_DEST_2 valid_for=(online_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_dest_state_3=ALTERNATE

  log_archive_config= 'dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB)
  fal_server=prifarsyncA,prifarsyncB,primary

- Standby Far Sync "A" (sbfarsyncA)

  log_archive_dest_2=" service="primary", ASYNC reopen=5 db_unique_name="primary" valid_for=(standby_logfile,all_roles)

  log_archive_dest_state_2=ENABLE

  log_archive_config= 'dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB)
  fal_server=standby

[5] http://docs.oracle.com/database/121/DGBKR/dbresource.htm#DGBKR3852

» Standby Far Sync “B” (sbfarsyncB)

log_archive_dest_2=” service="primary", ASYNC reopen=5 db_unique_name="primary" valid_for=(standby_logfile,all_roles)

log_archive_dest_state_2=ENABLE

log_archive_config= ‘dg_config=(primary,prifarsyncA, prifarsyncB, standby, sbfarsyncA, sbfarsyncB) fal_server=standby

*The above parameters are relevant for this configuration when set by SQL*Plus and are provided to make all the details of the configuration clear to the reader, however Oracle recommends the simpler approach of configuring Far Sync using the Data Guard Broker. Steps for configuring Far Sync with Data Guard Broker are listed in Appendix C*

For Oracle RDBMS Version 12.1 set the parameter ”_redo_transport_stall_time”=60 in all instances within the configuration for best return to synchronization from a node outage with an ALTERNATE Far Sync configuration.

Far Sync using Oracle Real Application Clusters – Oracle RAC

A Far Sync instance can also be placed on an Oracle RAC cluster. In this configuration Far Sync is only active on one server at a time while other servers provide automatic failover for HA. See Appendix A for steps to convert a single instance Far Sync to a RAC Far Sync.

The characteristics of this approach include:

» Lowest application brown-out when a Far Sync instance fails.

» Fastest resumption of zero data loss protection after Far Sync instance failure.

» By itself, this solution does not address cluster failure, configuration of an alternate destination is still required to maintain data protection should the cluster become unavailable.

Far Sync Instance Failure in an Oracle RAC Cluster

Figure 8 shows the impact on application throughput and data protection should there be a failure of Far Sync instance receiving redo when all RAC nodes are still functional.

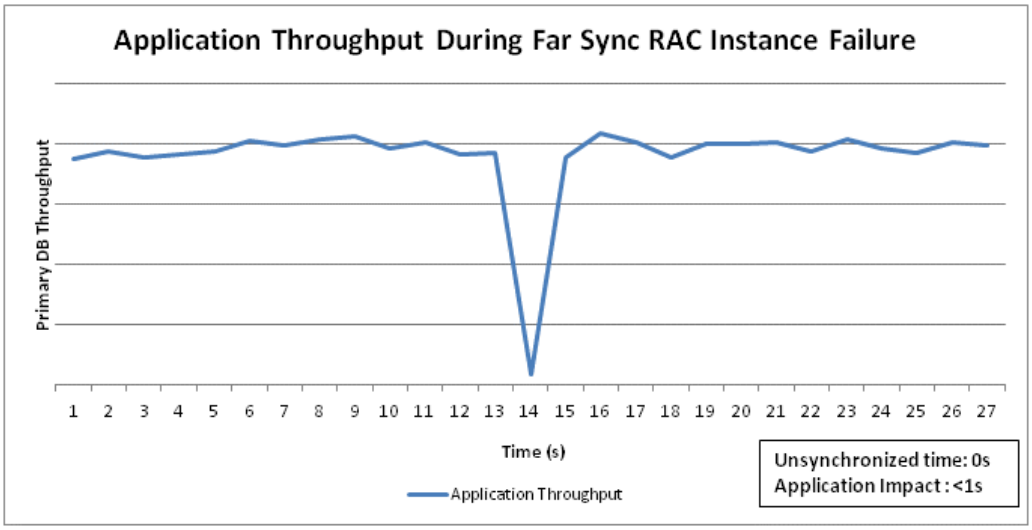

Figure 8 – Far Sync Instance Failure – all RAC nodes functional

In this use-case when the active Far Sync instance fails, Oracle RAC immediately detects the outage and automatically fails over redo transport to an already running Far Sync instance on one of the surviving RAC nodes (no alternate destination needs to be defined for this to occur). There is a very brief application brownout of less than one second to acknowledge and process the error notification during instance failover (net_timeout does not apply to this error state). The configuration remains at Maximum Availability protection level maintaining zero data loss protection throughout transition from one node of the cluster to the next – no re-transmission is necessary due to the new Far Sync instance being able to access existing SRLs on the cluster's shared storage. Fast detection and no interruption of zero data loss protection are substantial benefits of using Oracle RAC to host the Far Sync instance.

Node (Server) Failure in an Oracle RAC Cluster

Figure 9 shows the impact of an Oracle RAC node failure; an outage of the server on which the active Far Sync instance is running. Node failure incurs a brief brownout equal to Data Guard net_timeout causing the first dip in application throughput. Then a second brownout occurs while Data Guard redo transport re-establishes connection with the surviving node. Data loss potential is greater in this case because the node failure results in Data Guard entering a resynchronization state. Data Guard quickly resynchronizes the primary and standby and returns the configuration to a zero data loss protection level after the new connection is established. The time required to resynchronize the configuration will vary depending on how much redo needs to be transmitted to the surviving instance ('redo ship' in Figure 9).

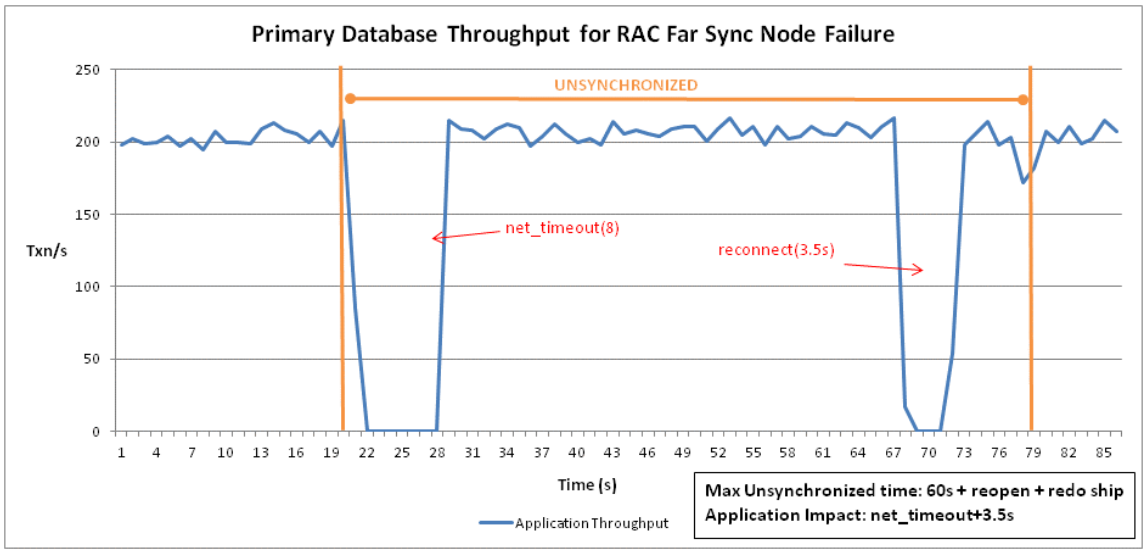

Figure 9 – Oracle RAC node (server) failure

The configuration details for the RAC use case are identical to the first example in this section: "HA using the Terminal Standby as an Alternate Destination" except for the additional steps in Appendix A to convert a single-instance Far Sync to Oracle RAC. Also note that the remote standby database can still be defined as an alternate

destination just as it is in the original example to provide data protection should the entire RAC cluster become unavailable.

For Oracle RDBMS Version 12.1 set the parameter "_redo_transport_stall_time"=60 in all instances within the configuration for best return to synchronization from a node outage with an ALTERNATE Far Sync configuration.

Choosing a Far Sync Deployment Topology

All configurations for Far Sync HA perform equally with regard to receiving and sending redo. The determination of which configuration to choose should be based on application tolerance to data loss should there be a double failure and to the brownout period of each configuration during different failure scenarios.

- A RAC Far Sync provides the lowest brownout and best protection. It also requires an Oracle RAC license

  The most critical applications are well served by a pair of RAC Far Sync instances, each configured as an alternate for the other and deployed at different locations. This provides the most robust HA and data protection (during instance, node, cluster and site outages).
- An alternate Far Sync instance in a non-RAC environment will have a slightly longer application brownout and greater potential data loss should a second outage impact the primary database while transport transitions from one Far Sync instance to the other. This option does not require an Oracle RAC license.

  Applications where data protection is critical but where cost is an important consideration are best served by deploying a pair of single node Far Sync instances, each as an alternate for the other.
- Use of the terminal standby as an alternate destination requires accepting that the configuration will run in asynchronous mode during the entire period required to resolve the Far Sync instance outage.

  The advantage of this approach is that it requires no additional hardware or software to deploy or manage. Applications that can tolerate increased data loss potential during a Far Sync instance outage and where low cost is the main consideration are best served by configuring the terminal standby as an alternate location using ASYNC redo transport.
- A Far Sync hub is an efficient way of consolidating Far Sync instances for multiple Data Guard configurations on a single physical host.

  Cloud deployments that include a zero data loss service level category can deploy a Far Sync hub to efficiently consolidate Far Sync instances for multiple zero data loss configuration on a single physical machine or cluster.

Data Guard Fast Sync

Fast Sync is a new Data Guard capability available with Oracle Database 12c. Fast Sync enables use of the destination parameter NOAFFIRM which specifies that the standby acknowledge receipt of redo without waiting for the write to the standby redo log file to complete. Fast Sync can improve application response time in a SYNC configuration by removing remote I/O from the total round trip time. It also prevents fluctuations and outliers in standby I/O performance from impacting application response time. Fast Sync can make it practical to increase the distance between the primary and any SYNC destination (a Far Sync instance or a standby database) to provide greater geographic protection.

Fast Sync is a feature of Data Guard that is included with every Oracle Database Enterprise Edition license. Fast Sync does not require an Active Data Guard license.

Oracle performance testing on Exadata demonstrated that primary database throughput increased by 4% when Fast Sync was configured (see Figure 10). Ironically, the fast I/O on an Exadata system using flashlogs results in more

modest performance advantages. A more substantial performance advantage is seen on primary databases deployed on systems having slower I/O. In a performance test with a Fast Sync instance on a virtual machine and NAS storage, for example, using Fast Sync resulted in a 12% increase in primary database throughput. This is due to the fact that Far Sync instance disk I/O is a larger percentage of total round-trip time in the virtual machine example.

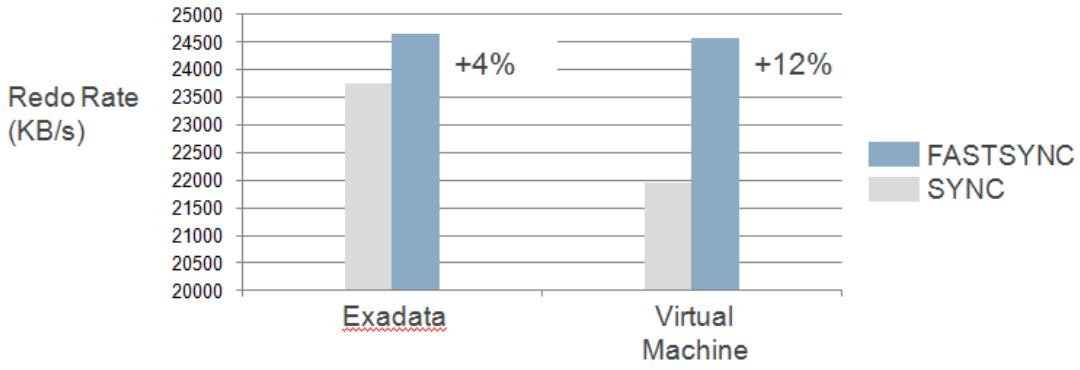

Figure 10 – Performance Comparison: FASTSYNC vs. SYNC

Data Guard Redo Transport Compression

Redo Transport Compression (RTC) can be CPU intensive thus it is beneficial to offload RTC to the Far Sync instance and save CPU cycles on the primary database host.

Previous MAA tests have shown that CPU utilization of each ASYNC transport process can increase to as much as 80% of a CPU while RTC is enabled during active processing. Note that in an Oracle RAC primary there is one ASYNC transport process per redo thread for each remote destination. The use of redo transport compression on a Far Sync instance is an important element to sizing the server where the Far Sync instance will reside.

In order to estimate the compression ratio for a given workload, set log_archive_trace=4 on the source(Far Sync) instance and monitor the tt0n process trace files for output similar to this example:

RTC: Compression ratio for this I/O is 80%

RTC: actual compressed bytes# 99

Additionally, you can confirm the savings in bandwidth used by the network interface of the Far Sync instance (likely eth0), before and after enabling compression using a UNIX ‘sar’ command similar to the following:

sar -n DEV <seconds> <# samples>

Active Data Guard and Data Guard do not include a license for redo transport compression. The primary database and all standby databases that receive compressed redo must be licensed for the Oracle Advanced Compression Option (ACO). The Far Sync instance does not require a separate ACO license as long as primary and standby databases are licensed appropriately for ACO.

Data Guard Redo Transport Encryption

In cases where redo transport encryption is enabled, encryption must be enabled between the Primary and Far Sync as well as between the Far Sync and terminal standby. This is due to the redo being unencrypted when taken off the wire at the Far Sync instance.

Starting with Oracle Database 11g Release 2, network encryption (native network encryption and SSL/TLS) and strong authentication services (Kerberos, PKI, and RADIUS) are no longer part of the Oracle Advanced Security Option and are available in all licensed editions of all supported releases of the Oracle database. See My Oracle Support Note 749947.1 for details on enabling transport encryption.

Conclusion

Active Data Guard Far Sync, a new capability with Oracle Database 12c, eliminates compromise by extending zero data loss protection to a replica database located at any distance from the primary database. Active Data Guard Far Sync accomplishes this with minimal expense or complexity compared to other extended distance data protection and availability solutions. The presence of a Far Sync instance in a Data Guard configuration is transparent to operation during switchover or failover, the administrator uses the same commands used for any Data Guard configuration. Far Sync requires nothing new to learn or any additional procedures to perform a zero data loss failover across a wide area network (WAN).

A Far Sync instance also offloads the primary of any overhead of resolving gaps in archived logs received by the remote standby database (e.g. following network or standby database outages) and can conserve WAN bandwidth by performing redo transport compression without impacting primary database performance (off host compression).

Far Sync has immediate appeal to any enterprise seeking zero data loss protection and availability over long distances. Far Sync also benefits users who have never seriously considered zero data loss protection due to performance concerns. Upgrading data protection using Far Sync enables anyone currently using an asynchronous replication strategy to eliminate the uncertainty and administrative burden that accompanies the need to reconcile data loss after a failover has occurred. Implementing off-host redo transport compression can conserve network bandwidth and reduce cost.

APPENDIX A: Converting a Single Instance Far Sync to RAC

For details of regarding creating and configuring an Oracle Data Guard environment with a single node Far Sync instance please refer to the Data Guard Concepts and Administration Guide[6].

To convert a single instance far sync to RAC:

1) Create a pfile from spfile and add the following parameters to convert the single instance far sync to RAC:

   cluster_database=true

   remote_listener=<*SCAN Listener name*>*:1521*

   For each instance in the RAC add an instance specific entry for the following parameters:

   instance_number

   thread

   local_listener

   A 2-node RAC example :

   farsync1.thread=1
   farsync2.thread=2
   farsync1.instance_number=1
   farsync2.instance_number=2
   farsync1.local_listener=(ADDRESS=(PROTOCOL=tcp)(HOST=<1stnodeIP>)(PORT=1521))
   farsync2.local_listener=(ADDRESS=(PROTOCOL=tcp)(HOST=<2ndnodeIP>)(PORT=1521))
   **.remote_listener=<SCAN Listener name>:1521*
   **.cluster_database=true*

2) Startup one instance with new pfile

3) Create spfile='+DATA/farsync/spfilefarsync.ora' from pfile='far_sync.pfile';

4) Copy password file to shared location(ASM diskgroup) in the cluster

5) Register the far sync with CRS:

   Example:

   srvctl add database -db farsync -oraclehome /u01/app/oracle/product/12.1/dbhome_1 -startoption mount -spfile '+DATA_FS/farsync/spfilefarsync.ora' -pwfile '+DATA/farsync/orapwfarsync'

   srvctl add instance -db farsync -i farsync1 –n <node1>

   srvctl add instance -db farsync -i farsync2 –n <node2>

6) Make appropriate changes to necessary tnsnames.ora entries on all sites to use the scan listener of the far sync so that either RAC instance may be used.

[6] http://docs.oracle.com/database/121/SBYDB/create_fs.htm#SBYDB5416

APPENDIX B: Creating a Far Sync Broker Configuration

This example describes steps to configure Oracle Data Guard Broker with a primary database with primary and alternate far sync and a single terminal standby database with primary and alternate far sync.

For additional detail regarding configuring Oracle Data Guard Broker please reference the Oracle documentation[7].

dgmgrl sys/oracle

DGMGRL> create configuration 'DGconfig' as primary database is ‘primary’ connect identifier is primary;
Configuration "DGconfig" created with primary database "primary"

DGMGRL> ADD FAR_SYNC prifarsyncA as connect identifier is prifarsyncA;
far sync instance " prifarsyncA " added

DGMGRL> ADD FAR_SYNC prifarsyncB as connect identifier is prifarsyncB;
far sync instance "dbmfsa" added

DGMGRL> ADD FAR_SYNC sbfarsyncA as connect identifier is sbfarsyncA;
far sync instance "sbfarsyncA" added

DGMGRL> ADD FAR_SYNC sbfarsyncB as connect identifier is sbfarsyncB;
far sync instance "sbfarsyncb" added

DGMGRL> add database standby as connect identifier is standby;
Database "standby" added

DGMGRL> edit far_sync prifarsyncA set property RedoRoutes='(primary :standby ASYNC)';
Property "redoroutes" updated

DGMGRL> edit far_sync prifarsyncB set property RedoRoutes='(primary : standby ASYNC)';
Property "redoroutes" updated

DGMGRL> edit far_sync sbfarsyncA set property RedoRoutes='(standby:primary ASYNC)';
Property "redoroutes" updated

DGMGRL> edit far_sync sbfarsyncB set property RedoRoutes='(standby:primary ASYNC)';
Property "redoroutes" updated

DGMGRL> edit far_sync prifarsyncA set property MaxFailure=1;
Property "maxfailure" updated

[7] http://docs.oracle.com/database/121/DGBKR/toc.htm

```
DGMGRL> edit far_sync prifarsyncB set property MaxFailure=1;
Property "maxfailure" updated

DGMGRL> edit far_sync sbfarsyncA set property MaxFailure=1;
Property "maxfailure" updated

DGMGRL> edit far_sync sbfarsyncB set property MaxFailure=1;
Property "maxfailure" updated

DGMGRL> edit far_sync prifarsyncA set property NetTimeout='8';
Property "nettimeout" updated

DGMGRL> edit far_sync prifarsyncB set property NetTimeout='8';
Property "nettimeout" updated

DGMGRL> edit far_sync sbfarsyncA set property NetTimeout='8';
Property "nettimeout" updated

DGMGRL> edit far_sync sbfarsyncB set property NetTimeout='8';
Property "nettimeout" updated

DGMGRL> edit far_sync prifarsyncA set property reopensecs=5;
Property "reopensecs" updated

DGMGRL> edit far_sync prifarsyncB set property reopensecs=5;
Property "reopensecs" updated

DGMGRL>  edit far_sync sbfarsyncA set property reopensecs=5;
Property "reopensecs" updated

DGMGRL> edit far_sync sbfarsyncB set property reopensecs=5;
Property "reopensecs" updated

DGMGRL> edit far_sync prifarsyncA set property LogXptMode='FASTSYNC';
Property "logxptmode" updated

DGMGRL> edit far_sync prifarsyncB set property LogXptMode='FASTSYNC';
Property "logxptmode" updated

DGMGRL> edit far_sync sbfarsyncA set property LogXptMode='FASTSYNC';
Property "logxptmode" updated

DGMGRL> edit far_sync sbfarsyncB set property LogXptMode='FASTSYNC';
Property "logxptmode" updated
```

```
DGMGRL> edit database primary set property RedoRoutes='(LOCAL : prifarsyncA FASTSYNC ALT=( prifarsyncB
FASTSYNC FALLBACK))';
Property "redoroutes" updated

DGMGRL>  edit database standby set property RedoRoutes='(LOCAL : sbfarsyncA FASTSYNC ALT=( sbfarsyncB
FASTSYNC FALLBACK))';
Property "redoroutes" updated

DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MaxAvailability;
Succeeded.

DGMGRL> enable configuration
Enabled.

DGMGRL> show configuration

Configuration - DGconfig

  Protection Mode: MaxAvailability
  Members:
  primary      - Primary database
    prifarsyncA    - Far sync instance
      standby   - Physical standby database

  Members Not Receiving Redo:
  prifarsyncB   - Far sync instance (alternate of prifarsyncA)
  sbfarsyncA  - Far sync instance
  sbfarsyncB - Far sync instance

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS   (status updated 15 seconds ago)
```

APPENDIX C: Oracle Net Configuration

Far Sync configurations should minimally have tnsnames.ora entries like the example below for each site as follows. It may however be easier to simply create an entry for all elements at all sites

For any site which contains RAC, all nodes must have the same TNS entries. For entries which point to a RAC site use the scan listener for HOST

Primary site: TNS entries for- primary, standby, all primary far sync instances(for RAC far sync use the scan listener

Each Primary Far Sync site: TNS entries for - primary, standby

Standby site: TNS entries for - primary, standby all standby far sync instances(for RAC far sync use the scan listener)

Each Standby Far Sync site: TNS entries for - primary, standby

```
PRIMARY =
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = <hostname>)(PORT = 1521))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = primary)
   )
 )
```

For an entry pointing to a RAC site:

```
PRIMARY_RAC =
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = <scan listener name>)(PORT = 1521))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = primary)
   )
 )
```

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