

数据库 DG 配置、管理手册

目录

1	Oracle Data Guard 介绍.....	8
1.1	Data Guard 简介.....	8
1.2	Standby Database 类型.....	9
1.2.1	Physical standby database.....	9
1.2.2	Logical standby database.....	9
1.2.3	Snapshot standby database.....	9
1.2.4	Active Data Guard (11G 新特性).....	10
1.3	Data Guard Services 类型.....	10
1.3.1	Redo transport services.....	10
1.3.2	Apply services.....	11
1.3.3	Role management services.....	11
1.4	保护模式.....	11
1.4.1	Maximum protection.....	11
1.4.2	Maximum availability.....	12
1.4.3	Maximum performance.....	12
1.5	Switchover 与 Failover.....	12
1.5.1	Switchover.....	12
1.5.2	Failover.....	12
1.6	Data Guard Broker.....	13
1.7	Data Guard 配置建议.....	14
1.7.1	数据库配置建议.....	14
1.7.2	网络相关参数调整.....	15
1.7.3	主库 RMAN 配置建议.....	18
1.7.4	备库 RMAN 配置建议.....	19
1.7.5	备份计划.....	19
1.8	Data Guard 配置选项.....	20
1.8.1	保护模式.....	20
1.8.2	重做传输.....	20
1.8.3	应用方式.....	20
1.8.4	角色转换.....	20
2	Data Guard 架构.....	21

2.1	Data Guard 核心进程.....	21
2.1.1	主数据库核心进程.....	21
2.1.2	备用数据库核心进程.....	22
2.1.3	Data Guard Broker.....	22
2.2	Automatic Gap Detection and Resolution.....	22
2.3	核心参数.....	23
2.3.1	ARCHIVE_LAG_TARGET.....	23
2.3.2	DB_FILE_NAME_CONVERT/LOG_FILE_NAME_CONVERT.....	24
2.3.3	DB_LOST_WRITE_PROTECT.....	24
2.3.4	DB_NAME.....	25
2.3.5	DB_UNIQUE_NAME.....	25
2.3.6	FAL_CLIENT/FAL_SERVER.....	25
2.3.7	LOG_ARCHIVE_CONFIG.....	26
2.3.8	LOG_ARCHIVE_DEST_n.....	26
2.3.9	LOG_ARCHIVE_DEST_STATE_n.....	29
2.3.10	STANDBY_FILE_MANAGEMENT.....	29
2.3.11	相关参数检查脚本.....	30
3	物理备用数据库.....	31
3.1	创建物理备用数据库 (使用 RMAN DUPLICATE).....	31
3.1.1	开启数据库 FORCE LOGGING (Primary Database).....	31
3.1.2	启动归档模式 (Primary Database).....	31
3.1.3	创建 STANDBY LOG (Primary Database).....	32
3.1.4	设置 Primary Database 参数 (Primary Database).....	33
3.1.5	配置 Oracle Net Service Name (Primary/Standby Database).....	34
3.1.6	Standby Database Listener (Standby Database).....	35
3.1.7	复制 Primary Database 口令文件 (Primary Database).....	36
3.1.8	创建 Physical Standby Database 参数文件 (Standby Database).....	36
3.1.9	创建 Physical Standby Database 所需目录 (Standby Database).....	37
3.1.10	启动 Physical Standby Database (Standby Database).....	37
3.1.11	使用 RMAN 执行 DUPLICATE (Primary Database).....	37
3.1.12	启用 Real-Time Apply (Standby Database).....	39
3.1.13	防止 Primary Database 数据库冲突 (Primary/Standby Database)	

3.2	物理备用数据库监控.....	39
3.2.1	查看备用数据库日志接收情况.....	39
3.3	物理备用数据库管理.....	40
3.3.1	物理备用数据库启停.....	40
3.3.2	打开物理备用数据库.....	40
3.3.3	需要手工维护的主数据库变更.....	43
3.4	转换为快照备用数据库.....	44
3.4.1	停止日志应用.....	45
3.4.2	确保数据库处于 MOUNT 状态，而不是 OPEN.....	45
3.4.3	确保配置了闪回区.....	45
3.4.4	执行转换操作.....	45
3.4.5	执行转换回退操作.....	45
4	Logical Standby Database.....	46
5	Data Guard 保护模式.....	47
5.1	保护模式类型.....	47
5.1.1	Maximum Availability.....	47
5.1.2	Maximum Performance.....	47
5.1.3	Maximum Protection.....	48
5.2	修改保护模式.....	48
5.2.1	确认 REDO 传输参数.....	48
5.2.2	修改保护模式.....	49
5.2.3	查看数据库保护模式.....	49
6	Redo 传输服务.....	50
6.1	Redo 传输的安全.....	50
6.2	发送 Redo Data.....	51
6.3	接收 Redo Data.....	51
6.4	级联 Redo 传输.....	51
6.5	监控 Redo 传输服务.....	52
6.5.1	查看当前最大归档日志.....	52
6.5.2	查看各日志传输目标状态.....	53
6.5.3	查询归档目标中不一致的日志.....	53
6.5.4	跟踪日志传输.....	53

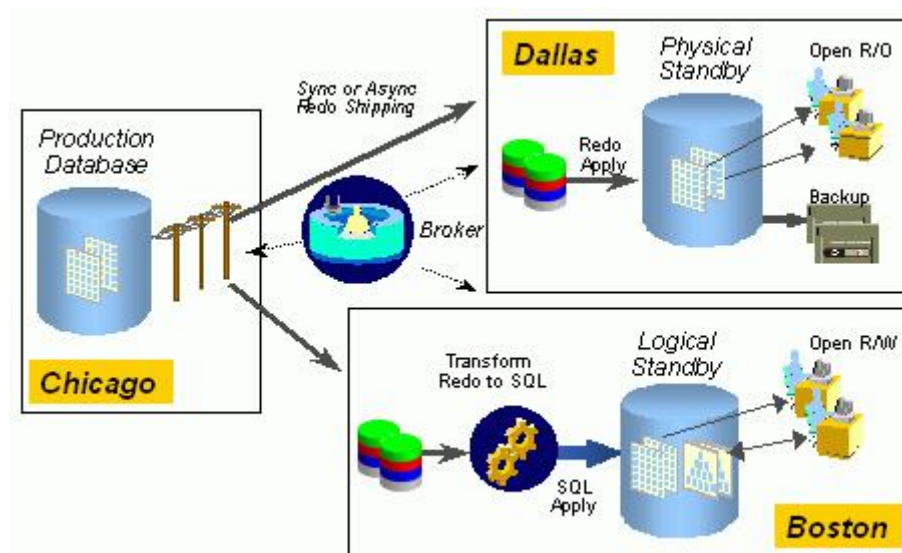
6.5.5	查看同步 Redo 传输响应时间.....	54
6.6	Redo Gap 检测和处理.....	54
6.6.1	手工处理物理备用数据库 Redo Gap.....	55
6.6.2	手工处理逻辑备用数据库 Redo Gap.....	55
6.7	Redo 传输等待事件.....	56
7	Apply Services.....	57
7.1	Apply Services 类型.....	57
7.2	Apply Services 配置.....	58
7.2.1	Real-Time Apply.....	58
7.2.2	Time Delay Apply.....	58
7.3	物理备用数据库的日志应用.....	59
7.3.1	启动日志应用.....	59
7.3.2	停止日志应用.....	59
7.4	逻辑备用数据库的日志应用.....	59
7.4.1	启动日志应用.....	59
7.4.2	停止日志应用.....	59
8	角色转换.....	60
8.1	物理备用数据库 Switchover.....	60
8.1.1	切换前的准备.....	60
8.1.2	选择角色转换的备用数据库.....	61
8.1.3	执行切换.....	61
8.2	物理备用数据库 Failover.....	64
8.2.1	执行故障切换.....	64
8.3	逻辑备用数据库 Switchover.....	66
8.4	逻辑备用数据库 Failover.....	66
8.5	角色转换触发器.....	67
8.6	闪回数据库.....	67
9	物理备库和快照备库管理.....	68
9.1	开启和关闭物理备库.....	68
9.1.1	开启物理备库.....	68
9.1.2	关闭物理备库.....	68
9.2	开启备库.....	68

9.2.1	实时查询.....	68
10	DG 管理与监控.....	70
10.1	常用数据字典.....	70
10.1.1	V\$DATABASE.....	70
10.1.2	V\$MANAGED_STANDBY.....	71
10.1.3	V\$ARCHIVED_LOG.....	71
10.1.4	V\$LOG_HISTORY.....	71
10.1.5	V\$ARCHIVE_DEST.....	72
10.1.6	V\$ARCHIVE_GAP.....	72
10.1.7	V\$DATAGUARD_CONFIG.....	72
10.1.8	V\$DATAGUARD_STATUS.....	72
10.1.9	V\$DATAGUARD_STATS.....	72
10.1.10	V\$STANDBY_LOG.....	72
10.1.11	V\$REDO_DEST_RESP_HISTOGRAM.....	72
10.2	Data Guard 启停.....	73
10.2.1	启动.....	73
10.2.2	停止.....	73
11	DG 应用案例.....	74
11.1	RMAN 备份恢复.....	74
11.2	级联 DG.....	74
11.2.1	环境信息规划.....	74
11.2.2	Cascading Standby Database 配置要求.....	75
11.2.3	主库准备工作.....	76
11.2.4	配置数据库静态监听.....	80
11.2.5	数据库参数配置.....	82
11.2.6	复制 Primary Database 口令文件 (Primary Database).....	86
11.2.7	创建 Standby Database 所需目录.....	86
11.2.8	启动 Standby Database.....	86
11.2.9	使用 RMAN 执行 DUPLICATE (Primary Database).....	87
11.2.10	启用 Real-Time Apply (Standby Database).....	89
11.3	RAC+DG(单实例).....	89
11.3.1	环境信息规划.....	89

11.3.2	RAC 主库准备.....	90
11.3.3	物理备库准备.....	96
11.3.4	创建物理备库.....	98
11.3.5	打开备用数据库.....	100
11.3.6	RAC 主库 SWITCHOVER 单机备库.....	100
11.3.7	新单节点主库 SWITCHOVER 旧 RAC 备库.....	103
11.3.8	FAILOVER 测试.....	105
11.3.9	常见问题处理.....	105
11.4	RAC+DG(RAC).....	105
12	故障处理.....	106
12.1	Manual Gap Resolution.....	106
13	参考文献.....	108

1 Oracle Data Guard 介绍

1.1 Data Guard 简介



Oracle Data Guard 是针对企业数据库的最有效和最全面的数据可用性、数据保护和灾难恢复解决方案。它提供管理、监视和自动化软件基础架构来创建和维护一个或多个同步备用数据库，从而保护数据不受故障、灾难、错误和损坏的影响。

Data Guard 备用数据库可以位于距离生产数据中心几千英里外的远程灾难恢复网站上，也可以位于同一个城市、同一个校园，甚至同一个建筑物中。如果生产数据库由于计划中或计划外中断而变得不可用，**Data Guard** 可以将任意备用数据库切换到生产角色，从而使停机时间减到最少并防止数据丢失。

Data Guard 提供：

可靠性 — 最佳的数据保护和可用性。您始终了解备用数据库的状态，并且它可以非常迅速地（在几秒钟内）切换为主数据库角色。

更低的成本和复杂性 — **Data Guard** 完善的功能和丰富的管理界面是 **Oracle** 企业版的特性。

最大的投资回报 — 所有备用数据库在作为备用角色的同时都可以用于生产目的，从而消除了闲置资源。

一个 Data Guard 配置包括一个生产数据库（或主数据库）和最多九个备用数据库。Data Guard 配置中的数据库通过 Oracle Net 连接，地理上可能非常分散。对数据库所处位置没有限制，只要它们能互相通信就行。

1.2 Standby Database 类型

1.2.1 Physical standby database

物理备用数据库提供与主数据库在物理上完全相同的副本，磁盘上的数据库结构与主数据库在块级别上完全相同。数据库模式（包括索引）都是相同的。重做应用技术使用标准 Oracle 介质恢复技术在物理备用数据库上应用重做数据。除传统的 Data Guard 功能外，适用于 Oracle 数据库 11g 的 Active Data Guard 选项使物理备用数据库可以在应用来自主数据库的更新时开启只读功能。这使物理备用数据库可以减少主数据库在处理只读查询和报表时的负担。这也使得对备用数据库是否时刻与主数据库同步的验证变得简单。

1.2.2 Logical standby database

虽然数据的物理组织和结构可能不同，但逻辑备用数据库与生产数据库包含相同的逻辑信息。SQL 应用技术通过将主数据库接收到的重做数据转换成 SQL 语句然后在备用数据库上执行 SQL 语句，使逻辑备用数据库与主数据库保持同步。这样，在将 SQL 应用到逻辑备用数据库上的同时，可以开启逻辑备用数据库的读写功能并访问逻辑备用数据库来进行查询和报表操作。

由于逻辑备用数据库是使用 SQL 语句更新的，因此它的灵活性高于物理备用数据库。这样，逻辑备用数据库可以读/写模式打开，并可以执行需要对数据库进行读/写访问的其他任务（如添加只在备用数据库中存在的本地表，或执行需要读/写权限的报表或合计任务）。通过在 SQL 应用维护的表上创建额外的索引和物化视图，可以优化这些任务（注意，尽管数据库是以读/写方式打开的，但 SQL 应用不允许更改负责与主数据库进行同步的数据）。逻辑备用数据库可以承载多个数据库模式，用户可以使用不从主数据库进行更新的模式，对表执行普通的数据处理操作。

逻辑备用数据库对数据类型、表的类型以及 DDL 和 DML 操作的类型有一些限制。

1.2.3 Snapshot standby database

快照备用数据库是一种可完全更新的备用数据库，其创建是由物理备用数据库转换而来。[快照备用数据库从主数据库接收并归档重做数据，但不应用这些重做数据](#)。在快照备用数据库重新转换为物理备用数据库并删除所有对快照备用数据库的本地更新之后，才应用从主数据库接收的重做数据。

要从 SQL*Plus 创建快照备用数据库，只需在物理备用数据库上发出下列命令：

```
SQL> ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;
```

备用数据库转换为快照备用数据库后，将创建一个隐式有保证的恢复点，并启用闪回数据库功能。使用完快照备用数据库后，可将其重新转换为物理备用数据库，并使用下列命令将其与主数据库重新同步：

```
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY DATABASE;
```

Data Guard 隐式地将数据库闪回到有保证的恢复点，并自动应用自快照创建以来备用数据库归档的主数据库重做数据。这一过程结束后，有保证的恢复点即被删除。

1.2.4 Active Data Guard (11G 新特性)

使用 **Active Data Guard** 选件的实时查询特性，可以在重做应用打开（恢复处于活动状态）的情况下以只读方式打开物理备用数据库，从而允许对主数据库的最新副本运行查询。这一特性使用企业管理器网格控制 **11g** 启用，或以下列方式从命令行手动启用：

要在重做应用处于活动状态时打开某个备用数据库进行只读访问，首先使用下列语句取消重做应用：

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

然后，使用下列语句打开数据库进行只读访问：

```
SQL> ALTER DATABASE OPEN;
```

打开备用数据库后，使用下列命令重启重做应用。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

1.3 Data Guard Services 类型

1.3.1 Redo transport services

Data Guard 配置包含一个生产数据库（也叫做主数据库）以及最多 9 个备用数据库（11G 以后支持 30 个备库）。**Data Guard** 通过重做数据来保持主数据库和备用数据库的同步。当主数据库中发生事务时，则生成重做数据（恢复事务所需的信息）并将其写入本地重做日志文件中。**Data Guard** 重做传输服务用于将此重做数据以同步或异步方式传送至备用

站点。这种灵活性使得备用数据库可以位于距生产数据中心数千英里的远程灾难恢复站点，也可位于同一城市、同一校园、同一建筑物乃至同一机房内。

1.3.2 Apply services

重做数据被传送到备用站点后，**Data Guard** 应用服务提供两种不同的方法用于将数据应用于备用数据库：**Data Guard** 重做应用（物理备用）或 **Data Guard SQL** 应用（逻辑备用）。

1.3.2.1 重做应用，用于物理备用数据库

物理备用数据库具有磁盘上的数据库结构，该结构每块都与主数据库完全一样，使用 **Oracle** 介质恢复进行更新。

1.3.2.2 SQL 应用，用于逻辑备用数据库

逻辑备用数据库是一个独立的数据库，它与主数据库包含相同的数据，使用 **SQL** 语句进行更新。每种方法都有其不同的优势，用户可根据自己的需求灵活选择。本文后面部分将详细讨论其差异。

1.3.3 Role management services

使用 **Data Guard**，数据库的角色可以在确保该过程中没有数据丢失且停机时间最少的前提下从主数据库切换到备用数据库，反之亦然。共有两种角色转换 — 转换和故障切换。转换是主数据库和它的一个备用数据库间的角色调换。通常在主系统计划维护中执行。在转换期间，主数据库将从主数据库角色转换为一个备用角色，而备用数据库将转换为主数据库角色。转换无需重新创建任何数据库。故障切换是备用数据库在主数据库突然中断时切换为主数据库角色。当问题得到修复后，故障主数据库可以使用 **Oracle** 闪回数据库迅速恢复为新主数据库的备用数据库，大大减少了将配置恢复为受保护状态的时间和工作量。管理员可以选择手动执行故障切换，也可以将 **Data Guard** 配置为自动检测主数据库故障并自动执行向备用数据库的故障切换（无需人工干预）。

1.4 保护模式

1.4.1 Maximum protection

该模式提供最高级别的数据保护。数据从主数据库同步传输到备用数据库，而且仅当重做数据至少在该模式中配置的一个备用数据库上可用时，才执行传输。如果该模式中配置的

最后一个备用数据库不可用，主数据库上的处理将停止。甚至在多方出现故障的情况下，该模式也可以确保无数据丢失。

To provide this level of protection, [the redo data needed to recover a transaction must be written to both the online redo log and to the standby redo log on at least one synchronized standby database before the transaction commits.](#)(保证在线日志和 standby 日志写成功,而不是备库 commit 完成) To ensure that data loss cannot occur, the primary database will shut down, rather than continue processing transactions, if it cannot write its redo stream to at least one synchronized standby database.

1. 4. 2 Maximum availability

该模式和最大保护模式相似（包括零数据丢失）。然而，如果一个备用数据库不可用（例如，由网络连接问题导致），主数据库上的处理仍将继续。当错误修复后，备用数据库会自动与主数据库同步。该模式在单一故障时可以实现无数据丢失（例如，网络故障、主站点故障）。

1. 4. 3 Maximum performance

该模式对主数据库提供相对较少的数据保护，但是提供比最大可用性模式更佳的性能。在该模式中，在主数据库处理事务时，重做数据异步传入备用数据库。主数据库的提交操作在完成主数据库上的写入操作之前不等待备用数据库确认收到重做数据。

1. 5 Switchover 与 Failover

1. 5. 1 Switchover

转换用于减少计划内中断期间（例如，操作系统或硬件升级，或者滚动升级 Oracle 数据库软件和补丁集）的主数据库停机时间。

转换操作要求断开所有用户会话与主数据库的连接。之后，将主数据库转换为备用角色，然后再将备用数据库转换为主数据库角色。

```
DGMGRL> SWITCHOVER TO Chicago;
```

一旦启动，Data Guard 就自动运行实际的角色转换进程。在该进程中，没有任何数据丢失。

1. 5. 2 Failover

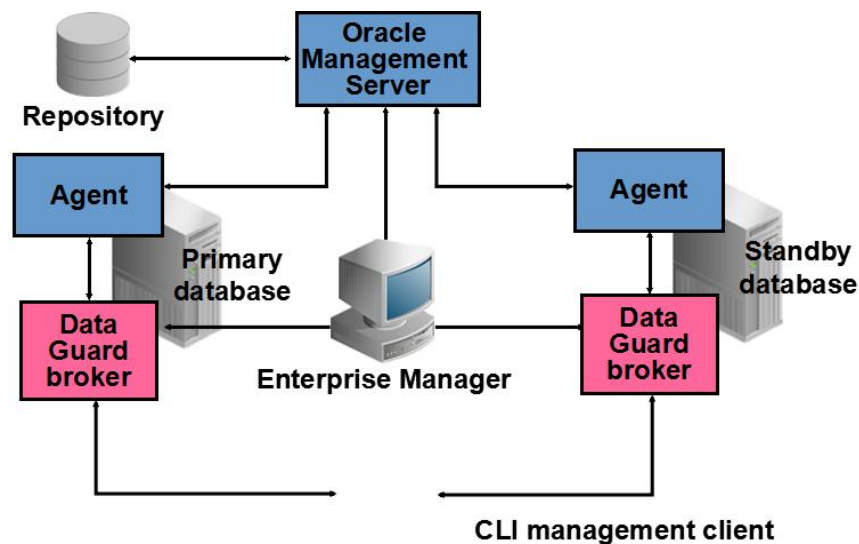
故障切换是在主数据库上出现计划外的故障时，将一个联机的备用数据库用作新的主数据库的操作。这种方法通过快速将备用数据库提升为主数据库角色来实现恢复时间目标，对

影响主数据库的事件进行诊断和解决的时候不会出现停机。故障切换操作不需要重启备用数据库。此外，只要原来的主数据库上的数据库文件未被更改且数据库可重启，则可以使用闪回数据库恢复原来的主数据库，并将其重新同步为新的主数据库的一个备用数据库 — 不需要从备份恢复。

DGMGRL> FAILOVER TO Chicago;

如果 Data Guard 运行在最大保护模式或最高可用性模式下，且在故障切换时已同步目标备用数据库，手动故障切换操作可以确保不会丢失数据。在最高性能模式下，如果主数据库中仍然有一些在进行故障切换时尚未发送到备用数据库的重做数据，该数据可能会丢失。

1.6 Data Guard Broker



Oracle Data Guard Broker 是一个分布式管理框架，它不但自动化了 Data Guard 配置的创建、维护和监视，并对这些操作进行统一管理。可以通过 Oracle 企业管理器（它使用 Broker）或 Broker 的专用命令行界面 (DGMGRL) 执行所有管理操作。Data Guard Broker 11g 还可以使用最大可用性或最佳性能模式将 Data Guard 配置为在数据库出现故障时自动切换。

1.7 Data Guard 配置建议

1.7.1 数据库配置建议

1.7.1.1 FRA (FAST RECOVERY AREA)

DB_RECOVERY_FILE_DEST = <mount point or Oracle ASM Disk Group>

DB_RECOVERY_FILE_DEST_SIZE = <disk space quota>

1.7.1.2 使用服务器参数文件

SPFILE 可以通过 RMAN 进行备份和恢复。

1.7.1.3 为主库、备库启用闪回数据库

可以将数据库还原到早期的时间点，而不需要还原整个数据库。

1.7.1.4 BCT (BLOCK CHANGE TRACKING)

使用 BLOCK CHANGE TRACKING，可以提高增量备份的效率。

```
SQL> SELECT FILENAME, STATUS, BYTES FROM V$BLOCK_CHANGE_TRACKING;
```

```
SQL> SELECT COUNT(*) FROM V$BACKUP_DATAFILE WHERE  
USED_CHANGE_TRACKING = 'NO'
```

1.7.1.5 ARCHIVE_LAG_TARGET

11G 使用 ADG 特性以后，该参数不再重要。

通过配置 ARCHIVE_LAG_TARGET 参数，可以定期的进行日志切换。参数单位为秒。

```
SQL> ALTER SYSTEM SET ARCHIVE_LAG_TARGET=1800 sid='*' scope=both;
```

1.7.1.6 LOG_ARCHIVE_MAX_PROCESSES

建议大于 5。

1.7.1.7 SRL (STANDBY REDO LOGS)

Standby redo logs are recommended for all Data Guard database configurations.

As best practice, you should create one more SRL group than the amount of ORL groups on the primary database. In addition, the SRLs should be of the same size. Lastly, SRLs should not be multiplexed.

1.7.1.8 FORCE LOGGING

You should set force logging on the primary database.

```
SQL> ALTER DATABASE FORCE LOGGING;
```

1.7.1.9 RTA (REAL-TIME APPLY)

Always enable RTA so that changes are applied as soon as the redo data is received.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT;
```

1.7.2 网络相关参数调整

1.7.2.1 SDU (SESSION DATA UNIT)

ORACLE 默认的网络数据传输单元为 8K，而日志传输的数据单元远大于 8K，因此将 SDU 设置为 32K，可以提高日志传输的效率。

调整 sqlnet.ora，会修改所有当前主机数据库的设置：

```
DEFAULT_SDU_SIZE=32767
```

通过 tnsnames.ora 设置 SDU：

```
Sample.com =  
(DESCRIPTION =  
  (SDU=32767)  
  (ADDRESS_LIST =  
    (ADDRESS =(PROTOCOL = TCP)(Host = Node1)(Port = 1521)  
  )  
)
```

```
(CONNECT_DATA = (SID = ORCL))  
)
```

通过 listener.ora 设置 SDU:

```
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = sqlaudit)(PORT = 1521))  
    )  
  )  
  
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC =  
      (SDU=32767)  
      (SID_NAME = ORCL)  
    )  
  )
```

1.7.2.2 BDP (BANDWIDTH-DELAY PRODUCT)

BDP 参数控制发送和接收缓存的大小。计算方式: $\text{Bandwidth} * \text{Latency} * 3$

通过 `ping -c 2 -s 64000 target_host` 检查网络的延迟。

```
[oracle@rac11g1 ~]$ ping -c 2 -s 64000 rac11g2
```

```
PING rac11g2 (192.168.56.130) 64000(64028) bytes of data.  
64008 bytes from rac11g2 (192.168.56.130): icmp_seq=1 ttl=64 time=7.46 ms  
64008 bytes from rac11g2 (192.168.56.130): icmp_seq=2 ttl=64 time=7.10 ms  
  
--- rac11g2 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1000ms  
rtt min/avg/max/mdev = 7.104/7.286/7.469/0.201 ms
```

该千兆网络的发送缓存可以设置为:

$(1000 * 1000 * 1000) * 8 / 1000 * 3 = 375000$ (字节)

➤ **TNSNAMES.ORA**

```
VISK_STDBY =  
  (DESCRIPTION =  
    (SDU=32767)  
    (SEND_BUF_SIZE=2557500)  
    (RECV_BUF_SIZE=2557500)  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = rac562-vip)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = VISK_DR)  
    )  
  )
```

➤ **LISTENER.ORA**

```
LISTENER =  
(DESCRIPTION_LIST =  
  (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = rac561-vip) (PORT =  
1521) (SEND_BUF_SIZE=2557500) (RECV_BUF_SIZE=2557500)))  
)
```

➤ **操作系统参数修改**

```
net.ipv4.tcp_wmem = 4096 16384 4194304
```

```
net.ipv4.tcp_rmem = 4096 87380 4194304
```

1.7.2.3 NETWORK QUEUE SIZE

In the world of Linux, you have two network queues, the interface transmit queue, and a network receive queue that should be adjusted for better performance.

➤ **发送队列**

```
[root@rac11g1 ~]# ifconfig eth0 txqueuelen 10000
```

➤ 接收队列

```
[root@rac11g1 ~]# sysctl -a | grep netdev_max_backlog
net.core.netdev_max_backlog = 1000
```

```
[root@rac11g1 ~]# echo "sys.net.core.netdev_max_backlog=20000" >> /etc/sysctl.conf
[root@rac11g1 ~]# sysctl -p
```

1.7.2.4 DISABLE TCP NAGLE ALGORITHM

To prevent delays in buffer flushing in the TCP protocol stack, disable the TCP Nagle algorithm by setting TCP_NODELAY to YES (the default value) in the SQLNET.ORA file on both the primary and standby systems.

```
TCP_NODELAY=YES
```

1.7.2.5 NTP 配置

You should enable NTP with the `-x` option to allow for gradual time changes, also referred to as slewing.

Drop root to id 'ntp:ntp' by default.

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

1.7.3 主库 RMAN 配置建议

➤ 备份保留策略

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <n> DAYS;

通过命令删除不再需要的备份

DELETE OBSOLETE;

➤ 归档日志删除策略

--传输到所有目标

CONFIGURE ARCHIVELOG DELETION POLICY TO SHIPPED TO ALL STANDBY;

--应用到所有目标

```
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

➤ 为数据库配置连接串

```
CONFIGURE DB_UNIQUE_NAME BOSTON CONNECT IDENTIFIER 'boston_conn_str';
```

[boston_conn_str](#) 是 TNS 中的配置串信息。

通过命令查看备库信息

```
RMAN> LIST DB_UNIQUE_NAME OF DATABASE;
```

List of Databases

DB Key	DB Name	DB ID	Database Role	Db_unique_name
2	ACCT	3830711767	PHYSICAL STANDBY	ACCTPHY

1.7.4 备库 RMAN 配置建议

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

```
CONFIGURE BACKUP OPTIMIZATION ON;
```

```
CONFIGURE CHANNEL DEVICE TYPE SBT PARMS '<channel parameters>';
```

➤ 备库不做备份

```
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

1.7.5 备份计划

1.7.5.1 使用磁盘作为磁带备份的缓存

➤ 每日备份

```
RESYNC CATALOG FROM DB_UNIQUE_NAME ALL;
```

```
RECOVER COPY OF DATABASE WITH TAG 'OSS';
```

```
BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 1 FOR RECOVER OF COPY  
WITH TAG 'OSS' DATABASE;
```

```
BACKUP DEVICE TYPE SBT ARCHIVELOG ALL;
```

```
BACKUP BACKUPSET ALL;
```

```
DELETE ARCHIVELOG ALL;
```

➤ 每周备份

```
BACKUP RECOVERY FILES;
```

1.7.5.2 使用磁带备份

➤ 每日备份

```
RESYNC CATALOG FROM DB_UNIQUE_NAME ALL;  
BACKUP AS BACKUPSET INCREMENTAL LEVEL 1 DATABASE PLUS ARCHIVELOG;  
DELETE ARCHIVELOG ALL;
```

➤ 每周备份

```
BACKUP AS BACKUPSET INCREMENTAL LEVEL 0 DATABASE PLUS ARCHIVELOG;  
DELETE ARCHIVELOG ALL;
```

1.8 Data Guard 配置选项

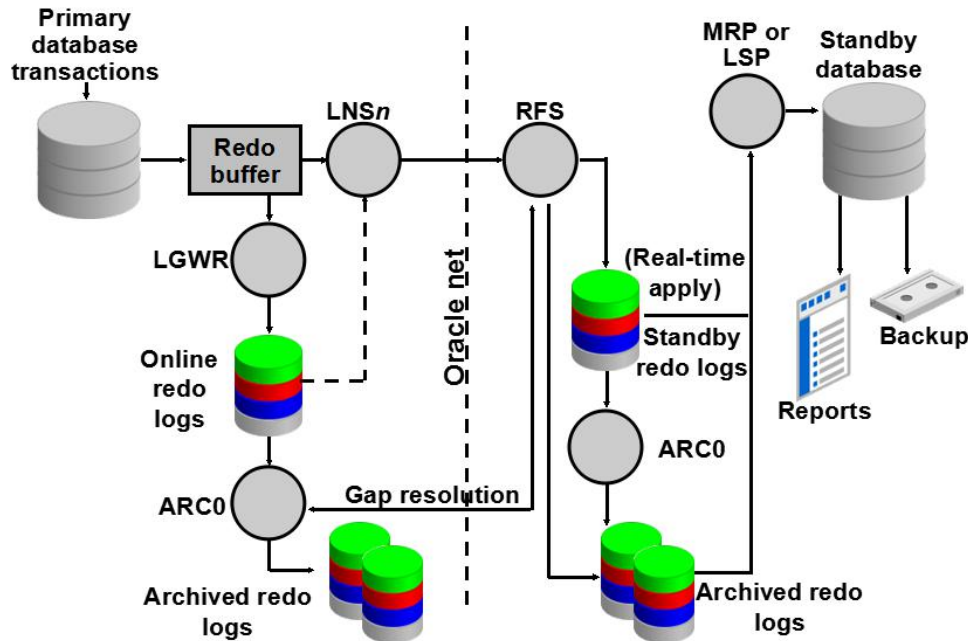
1.8.1 保护模式

1.8.2 重做传输

1.8.3 应用方式

1.8.4 角色转换

2 Data Guard 架构



2.1 Data Guard 核心进程

2.1.1 主数据库核心进程

➤ Log Writer Network Server (LNS)

Data Guard 在主数据库上使用一个叫做日志写入器网络服务器 (LNS) 的专门的后台进程来捕获日志写入器写入的重做数据，并以同步或异步方式将重做数据传输到备用数据库。LNS 进程将日志写入器与传输开销及网络中断隔离开来。

同步重做传输 (SYNC) 要求主数据库上的日志写入器等待 LNS 确认备用数据库已经接收重做数据并已将其写入备用重做日志，然后才能确认到客户端应用程序的提交。这确保了提交的所有事务都在磁盘上，并在备用位置受到保护。

异步重做传输 (ASYNC) 不要求主数据库上的日志写入器等待备用数据库确认重做已被写入磁盘。确认到客户端应用程序的提交与重做传输是异步进行的。

➤ Log writer (LGWR)

➤ Archiver (ARCn)

2.1.2 备用数据库核心进程

➤ Remote file server (RFS)

在 Data Guard 重做传输服务的接收方，Data Guard 在使用备用数据库上的一个或多个远程文件服务器 (RFS) 进程来接收重做数据，并将其写入一个名为备用重做日志 (SRL) 的文件。

➤ Managed recovery (MRP)/Logical standby (LSP)

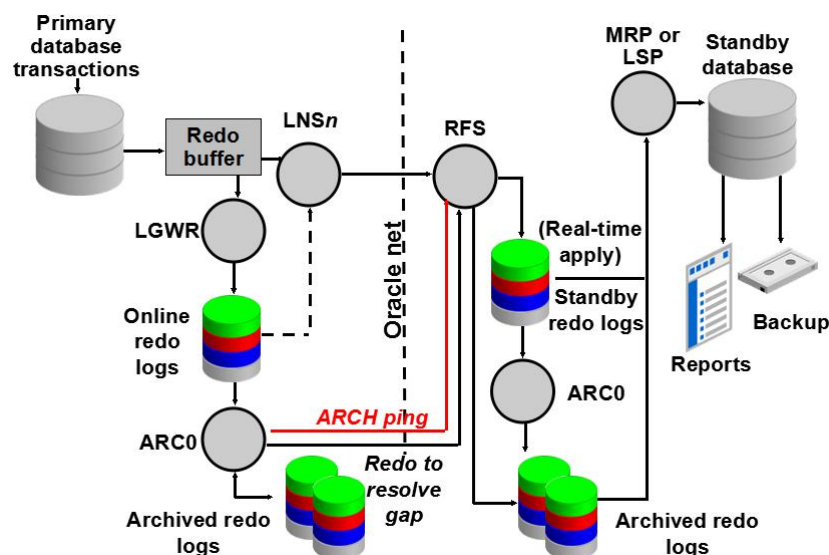
管理恢复进程 (MRP) 用于协调重做数据到物理备用数据库的应用，逻辑备用进程 (LSP) 用于协调经过 SQL 转换的重做数据到逻辑备用数据库的应用。

➤ Archiver (ARCn)

2.1.3 Data Guard Broker

如果启用了 Data Guard Broker，Data Guard 还使用 Data Guard Broker Monitor (DMON) 和其他进程将主数据库和备用数据库作为一个统一的配置进行管理和监视。

2.2 Automatic Gap Detection and Resolution



如果主数据库和备用数据库断开连接（网络故障或备用服务器故障），那么根据选择的保护模式，主数据库将继续处理事务并累积不能传输到备用数据库的重做数据积压，直到能建立一个新的网络连接为止（称为归档日志差异）。在这种状态下，Data Guard 持续监视

备用数据库状态、检测连接重建时间、并自动重新同步备用数据库与主数据库，以尽快将配置返回到受保护状态。始终使用归档器 (ARCn) 进程来传输解析差异所需的日志文件。有几种方法可用于加快解析差异的速度。

1. 使用 **Advanced Compression** 选项，可对为解析差异而传输的归档日志进行网络压缩，以便缩短传输时间，并最大程度地利用可用带宽。
2. 如果有大量归档日志需要传输，最多可启用 30 个 ARCn 进程。其中 29 个 ARCn 进程可传输到远程位置，一个 ARCn 进程始终专用于本地归档。
3. 如果积压的归档日志不多但日志的大小很大，最多可以配置 5 个 ARCn 进程并行传输一个日志文件的内容。

最后，如果备用应用进程检测到日志文件丢失，或其接收的归档日志中有任何缺陷或损坏，Data Guard 备用数据库将主动请求获取归档日志的新副本。这类请求由 ARCn 进程来完成。

2.3 核心参数

Parameter Name	Description
LOG_ARCHIVE_CONFIG	Specifies the unique database name for each database in the configuration
LOG_ARCHIVE_DEST_n	Controls redo transport services
LOG_ARCHIVE_DEST_STATE_n	Specifies the destination state
ARCHIVE_LAG_TARGET	Forces a log switch after the specified number of seconds
LOG_ARCHIVE_TRACE	Controls output generated by the archiver process
DB_FILE_NAME_CONVERT	Converts primary database file names
LOG_FILE_NAME_CONVERT	Converts primary database log file names
STANDBY_FILE_MANAGEMENT	Controls automatic standby file management

2.3.1 ARCHIVE_LAG_TARGET

控制数据库在指定的时间内自动切换日志。单位为秒。

```
SQL> ALTER SYSTEM SET archive_lag_target=1800;
```

2.3.2 DB_FILE_NAME_CONVERT/LOG_FILE_NAME_CONVERT

DB_FILE_NAME_CONVERT must be defined on standby databases that have different disk or directory structures from the primary.

Multiple pairs of file names can be listed in the DB_FILE_NAME_CONVERT parameter.

DB_FILE_NAME_CONVERT applies only to a physical standby database.

```
DB_FILE_NAME_CONVERT =('/oracle1/dba',  
                        '/ora1/stby_dba',  
                        '/oracle2/dba',  
                        '/ora2/stby_dba/')
```

If the standby database uses Oracle Managed Files (OMF), do not set the DB_FILE_NAME_CONVERT/LOG_FILE_NAME_CONVERT parameter.

2.3.3 DB_LOST_WRITE_PROTECT

Data Guard 的重要优势之一是能使用 Oracle 进程在重做数据应用于备用数据库之前对其进行验证。Data Guard 是一个松散耦合的体系结构，通过应用重做块保持备用数据库同步，完全不受主数据库中可能发生的数据文件损坏的影响。进行 SYNC 重做传输时，重做从主 SGA 发出，因此完全不受主数据库中的物理输入/输出损坏的影响。备用数据库中执行的软件代码路径也与主数据库完全不同 — 有效地将备用数据库与可能影响主数据库的软件错误隔离开来。

损坏检测检查在以下主要接口上进行：

重做传输过程中在主数据库上进行：LGWR、LNS、ARCH

重做应用过程中在备用数据库上进行：RFS、ARCH、MRP、DBWR

如果重做应用检测到备用数据库中有重做损坏，Data Guard 将在归档日志差异处理过程中重新获取有效日志，寄希望于原始的归档日志未受损坏。

当输入/输出子系统确认完成写入时，发生一个写入丢失问题，而实际上写入在持久存储中并未发生。随后进行块读取操作时，输入/输出子系统返回数据块的陈旧版本（可用于更新数据库的其他块），从而导致其损坏。若设置了 DB_LOST_WRITE_PROTECT 初始化参数，数据库将把缓冲区缓存块读取记录到重做日志中，此信息可用于检测写入丢失。写入丢失检测与 Data Guard 一起使用时最为有效。

在这种情况下，应在主数据库和备用数据库中都设置 **DB_LOST_WRITE_PROTECT** 参数。当某个备用数据库在管理恢复过程中应用重做时，它将读取相应的块并将 **SCN** 与重做日志中的 **SCN** 相对比：

1. 如果主数据库上的块 **SCN** 低于备用数据库，将检测主数据库上的写入丢失，并引发一个外部错误 (**ORA-752**)。推荐的修复主数据库上的写入丢失的步骤是在出现故障时切换到物理备用数据库，然后重新创建主数据库。
2. 如果 **SCN** 高于后者，将检测备用数据库上的写入丢失，并引发一个内部错误 (**ORA-600 3020**)。要修复备用数据库上的写入丢失，必须重新创建此备用数据库或受损的文件。

在上述两种情况下，备用数据库都将把故障原因写入警报日志和跟踪文件。

2.3.4 **DB_NAME**

2.3.5 **LOG_ARCHIVE_TRACE**

2.3.6 **DB_UNIQUE_NAME**

Data Guard 环境中的数据库，数据库通过 **DB_UNIQUE_NAME** 进行唯一区分。

2.3.7 **FAL_CLIENT/FAL_SERVER**

Fetch archive log (FAL):

Provides a client/server mechanism for resolving gaps detected in the range of archived redo logs that are generated at the primary database and received at the standby database.

Applicable for physical standby databases only.

Process is started only when needed, and shuts down as soon as it is finished.

2.3.7.1 **FAL_SERVER**

设置 **FAL_SERVER** 参数，此参数指定当日志传输出问题时，备库到哪里去找缺失的归档日志。它用在备库接收的到的重做日志间有 **gap** 的时候。这种情况会发生在日志传输出现中断时，如对备库需要维护而主库仍然正常运行，这时候，备库维护期间，没有日志传输过来，**gap** 就出现了，设置了这个参数，备库会主动去寻找那些缺失的日志，并要求主库进行传输。

2.3.7.2 FAL_CLIENT

FAL_CLIENT specifies the FAL (fetch archive log) client name that is used by the FAL service, configured through the FAL_SERVER initialization parameter, to refer to the FAL client. The value is an Oracle Net service name, which is assumed to be configured properly on the FAL server system to point to the FAL client (standby database).

该参数在 11g 中已经废弃。

2.3.8 LOG_ARCHIVE_CONFIG

```
LOG_ARCHIVE_CONFIG='DG_CONFIG=(pc00prmy,pc00sby1)'
```

```
SQL> SELECT * FROM V$DATAGUARD_CONFIG;
```

```
DB_UNIQUE_NAME
```

```
-----  
skyprim
```

```
skystand
```

2.3.9 LOG_ARCHIVE_DEST_n

该参数控制归档的位置。

```
LOG_ARCHIVE_DEST_1=
```

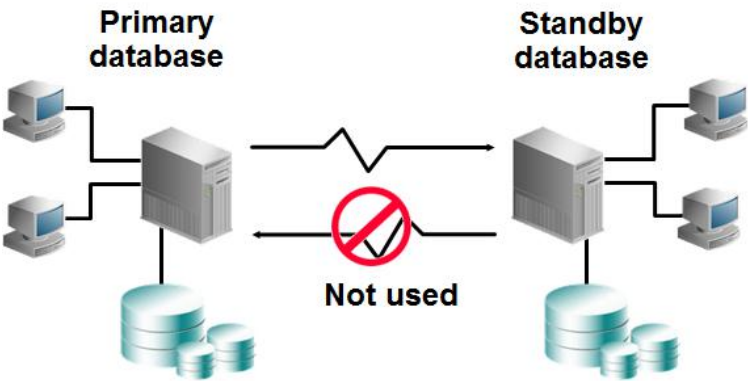
```
'SERVICE=pc00sby1
```

```
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
```

```
DB_UNIQUE_NAME=pc00sby1'
```

```
LOG_ARCHIVE_DEST_STATE_1=ENABLE
```

```
SELECT DEST_NAME, STATUS, DESTINATION FROM V$ARCHIVE_DEST;
```



■ Primary database

```
log_archive_dest_1 = 'service=pc00sby1 async valid_for=(online_logfile, primary_role)
db_unique_name=pc00sby1'
```

■ Standby database

```
log_archive_dest_1 = 'service=pc00prmy async valid_for=(online_logfile, primary_role)
db_unique_name=pc00prmy'
```

主要参数说明：

2.3.9.1 LOCATION and SERVICE

每个目的地必须指定的 **LOCATION** 或者是 **SERVICE** 属性来识别一个本地磁盘目录或一个远程数据库的目的地。

2.3.9.2 VALID_FOR

Combination	Primary	Physical	Logical
ONLINE_LOGFILE, PRIMARY_ROLE	Valid	Ignored	Ignored
ONLINE_LOGFILE, STANDBY_ROLE	Ignored	Ignored	Valid
ONLINE_LOGFILE, ALL_ROLES	Valid	Ignored	Valid
STANDBY_LOGFILE, STANDBY_ROLE	Ignored	Valid	Valid
STANDBY_LOGFILE, ALL_ROLES	Ignored	Valid	Valid
ALL_LOGFILES, PRIMARY_ROLE	Valid	Ignored	Ignored
ALL_LOGFILES, STANDBY_ROLE	Ignored	Valid	Valid

ALL_LOGFILES, ALL_ROLES	Valid	Valid	Valid
-------------------------	-------	-------	-------

配置错误，会出现如下错误：

ORA-16026: The parameter LOG_ARCHIVE_DEST_n contains an invalid attribute value

1. REDO_LOG_TYPE

ONLINE_LOGFILE: This destination is used only when archiving online redo log files.

STANDBY_LOGFILE: This destination is used only when archiving standby redo log files or receiving archive logs from another database.

ALL_LOGFILES: This destination is used when archiving either online or standby redo log files.

2. DATABASE_ROLE

PRIMARY_ROLE: This destination is used only when the database is in the primary database role.

STANDBY_ROLE: This destination is used only when the database is in the standby (logical or physical) role.

ALL_ROLES: This destination is used when the database is in either the primary or the standby (logical or physical) role.

2.3.9.3 SYNC and ASYNC

SYNC: Specifies that redo data generated by a transaction must have been received at a destination that has this attribute before the transaction can commit; otherwise, the destination is deemed to have failed. In a configuration with multiple SYNC destinations, the redo must be processed as described here for every SYNC destination.

ASYNC (default): Specifies that redo data generated by a transaction need not have been received at a destination that has this attribute before the transaction can commit

2.3.9.4 AFFIRM and NOAFFIRM

AFFIRM: 在主库 REDO 写进程进行之前，备库 REDO LOG 必须传输同步写完成。

NOAFFIRM: 主库 REDO 写进程不等待备库 REDO LOG 传输同步写完成。

如果 AFFIRM NOFFIRM 都没设置，那么 SYNC 默认是 AFFIRM，ASYNC 默认是 NOAFFIRM。

2.3.9.5 DELAY

一般设置延迟应用的需求都是基于容错方面的考虑，如 **Primary** 数据库端由于误操作，数据被意外修改或删除，只要 **Standby** 数据库尚未应用这些修改，你就可以快速从 **Standby** 数据库中恢复这部分数据。**DELAY** 属性影响备库应用 **REDO** 数据的延迟时间，单位分钟 默认值是 30 分钟，但不影响 **REDO** 数据的传输和归档到备库。

2.3.9.6 MANDATORY

指定了在线日志文件必须成功归档到目的地才可以重用。

2.3.9.7 NET_TIMEOUT

指定 **LGWR** 后台进程等待 **Redo** 传输目的地确认收到 **Redo** 数据的秒数，如果确认没有在 **NET_TIMEOUT** 秒内收到，一个错误被记录，同时到该目的地的 **Redo** 传输会话被中断。

2.3.10 LOG_ARCHIVE_DEST_STATE_n

Value	Description
ENABLE	Redo transport services can transmit redo data to this destination. This is the default.
DEFER	Redo transport services will not transmit redo data to this destination.
ALTERNATE	This destination will become enabled if communication to its associated destination fails.

2.3.11 STANDBY_FILE_MANAGEMENT

STANDBY_FILE_MANAGEMENT is used to maintain consistency when you add or delete a data file on the primary database.

When **STANDBY_FILE_MANAGEMENT** is set to **AUTO**, you cannot execute the following commands on the standby database:

ALTER DATABASE RENAME

ALTER DATABASE ADD/DROP LOGFILE [MEMBER]

ALTER DATABASE ADD/DROP STANDBY LOGFILE MEMBER

ALTER DATABASE CREATE DATAFILE AS ...

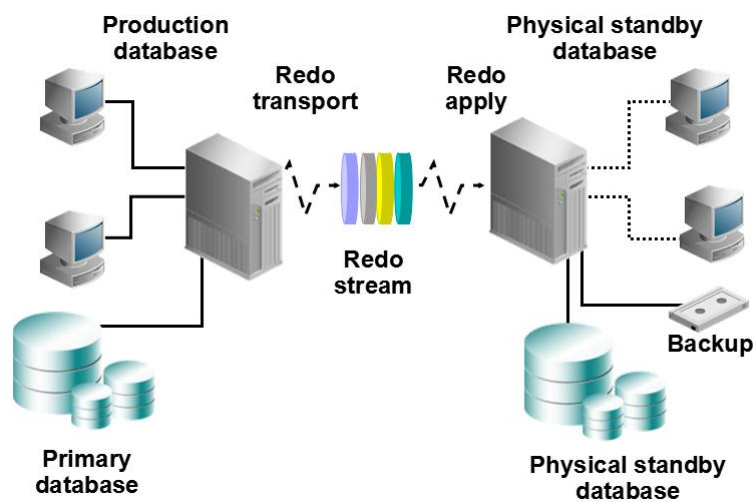
When you add a log file to the primary database and want to add it to the physical standby database as well (or when you drop a log file from the primary and want to drop it from the physical), you must do the following:

1. Set STANDBY_FILE_MANAGEMENT to MANUAL on the physical standby database.
2. Add the redo log files to (or drop them from) the primary database.
3. Add them to (or drop them from) the standby database.
4. Reset to AUTO afterward on the standby database.

2.3.12 相关参数检查脚本

```
SET LINESIZE 160;
COL NAME FOR A30;
COL VALUE FOR A80;
SELECT INST_ID, NAME, VALUE
FROM GV$PARAMETER
WHERE (NAME IN ('cluster_database',
                'db_name',
                'db_unique_name',
                'dg_broker_start',
                'dg_broker_config_file1',
                'dg_broker_config_file2',
                'fal_client',
                'fal_server',
                'archive_lag_target',
                'log_archive_max_processes',
                'log_archive_config',
                'log_archive_trace',
                'db_file_name_convert',
                'log_file_name_convert',
                'remote_login_password_file',
                'standby_file_management',
                'redo_transport_user'))
      OR (NAME LIKE 'log_archive_dest%' AND UPPER(VALUE) !=
UPPER('ENABLE'))
ORDER BY NAME, INST_ID;
```

3 物理备用数据库



3.1 创建物理备用数据库 (使用 RMAN DUPLICATE)

3.1.1 开启数据库 FORCE LOGGING (Primary Database)

SQL> `SELECT NAME, LOG_MODE, FORCE_LOGGING FROM V$DATABASE;`

NAME	LOG_MODE	FOR
SKY	NOARCHIVELOG	YES

开启:

SQL> `ALTER DATABASE FORCE LOGGING;`

关闭:

SQL> `ALTER DATABASE NO FORCE LOGGING;`

注意: This statement can take a considerable amount of time to complete, because it waits for all unlogged direct write I/O to finish.

3.1.2 启动归档模式 (Primary Database)

SQL> `ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=/oracle/app/oracle/oradata/archive/skyprim/';`

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

进行日志切换测试:

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

```
[oracle@rac11g1 ~]$ cd /oracle/app/oracle/oradata/archive/primary
```

```
[oracle@rac11g1 primary]$ ls -la *
```

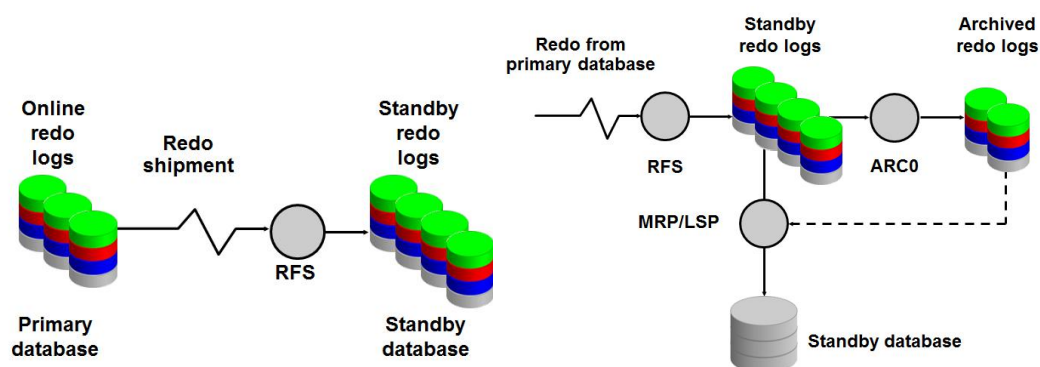
```
-rw-r----- 1 oracle asmadmin 38706688 Aug 17 17:31 1_9_887981944.dbf
```

3.1.3 创建 STANDBY LOG (Primary Database)

standby redo log 用于存储从其他数据库接收到的 redo 信息。

Redo received from another Oracle database via redo transport is written to the current standby redo log group by an RFS foreground process. When a log switch occurs on the redo source database, incoming redo is then written to the next standby redo log group, and the previously used standby redo log group is archived by an ARCn foreground process.

Each standby redo log file must be at least as large as the largest redo log file in the redo log of the redo source database (至少要有主数据库的日志大小). For administrative ease, Oracle recommends that all redo log files in the redo log at the redo source database and the standby redo log at a redo transport destination be of the same size (ORACLE 建议 redo 和 standby redo 大小一致, 数量要比 redo 多 1).



REDO 信息:

```
SQL> SELECT GROUP#, BYTES FROM V$LOG;
```

```
SQL> SELECT MEMBER FROM V$LOGFILE;
```

创建 STANDBY LOG:

```
ALTER DATABASE ADD STANDBY LOGFILE  
('/oracle/app/oracle/oradata/sky/standby1.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE  
('/oracle/app/oracle/oradata/sky/standby2.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE  
('/oracle/app/oracle/oradata/sky/standby3.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE  
('/oracle/app/oracle/oradata/sky/standby4.log') SIZE 50M;
```

检查 STANDBY LOG:

```
SQL> SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

```
SQL> SELECT GROUP#, TYPE, MEMBER FROM V$LOGFILE WHERE TYPE =  
'STANDBY';
```

3.1.4 设置 Primary Database 参数 (Primary Database)

Database	DB_UNIQUE_NAME	Oracle Net Service Name	DB_NAME
Primary	skyprim	skyprim	sky
Physical standby	skystand	skystand	

--Primary Role Initialization Parameters

```
DB_NAME=sky
```

```
DB_UNIQUE_NAME=skyprim
```

```
LOG_ARCHIVE_CONFIG='DG_CONFIG=(skyprim,skystand)'
```

```
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/app/oracle/oradata/archive/skyprim/
```

```
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=skyprim'
```

```
LOG_ARCHIVE_DEST_STATE_1=ENABLE
```

```
LOG_ARCHIVE_DEST_2='SERVICE=skystand ASYNC
```

```
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=skystand'
```

```
LOG_ARCHIVE_DEST_STATE_2=ENABLE
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE

--Standby Role Initialization Parameters
FAL_SERVER=skystand
DB_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/skystand/', '/oracle/app/oracle/oradata/sky/'
LOG_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/skystand/', '/oracle/app/oracle/oradata/sky/'
STANDBY_FILE_MANAGEMENT=AUTO
```

修改参数：

```
SQL> create pfile='/home/oracle/sky.pfile' from spfile;
```

```
SQL> shutdown immediate;
```

修改 **sky.pfile** 文件：

```
[oracle@rac11g1 ~]$ vi sky.pfile
```

```
DB_NAME=sky
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
```

注意部分参数原来的控制文件中已经存在。

重启数据库：

```
SQL> startup nomount pfile='/home/oracle/sky.pfile';
```

```
SQL> create spfile from pfile='/home/oracle/sky.pfile';
```

```
SQL> startup force;
```

3. 1. 5 配置 Oracle Net Service Name (Primary/Standby Database)

主数据库和备用数据库的 **tnsnames.ora** 文件都需要修改。

```
[oracle@rac11g1 ~]$ cd $ORACLE_HOME/network/admin
```

```
[oracle@rac11g1 admin]$ vi tnsnames.ora
```

```
skyprim =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.56.110)(PORT = 1521))
```

```
(CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = skyprim)
)
)

skystand =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.56.130)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = skystand)
    )
  )
)
```

```
[oracle@rac11g2 admin]$ tnsping skyprim
```

```
[oracle@rac11g2 admin]$ tnsping skystand
```

3. 1. 6 Standby Database Listener (Standby Database)

需要注意，11G 使用 ASM 的时候，监听文件在 grid 用户下。

```
[oracle@rac11g2 admin]$ su - grid
```

```
[grid@rac11g2 ~]$ cd $ORACLE_HOME/network/admin
```

```
[grid@rac11g2 admin]$ vi listener.ora
```

--新增内容

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = skystand)
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db_1)
      (SID_NAME = skystand)
    )
  )
)
```

```
[grid@rac11g2 admin]$ lsnrctl reload
```

```
[grid@rac11g2 admin]$ lsnrctl status
```

Note: This entry is needed because we start the instance in NOMOUNT mode.

3. 1. 7 复制 Primary Database 口令文件 (Primary Database)

```
[oracle@rac11g1 ~]$ cd $ORACLE_HOME/dbs
[oracle@rac11g1 dbs]$ scp orapwsky
192.168.56.130:/oracle/app/oracle/product/11.2.0/db_1/dbs
```

注意: Whenever you grant or revoke the SYSDBA or SYSOPER privileges or change the login password of a user who has these privileges, you must replace the password file at each physical or snapshot standby database in the configuration with a fresh copy of the password file from the primary database.

3. 1. 8 创建 Physical Standby Database 参数文件 (Standby Database)

```
[oracle@rac11g1 ~]$ scp sky.pfile 192.168.56.130:/home/oracle
```

```
*.audit_file_dest='/oracle/app/oracle/admin/skystand/adump'
*.audit_trail='db'
*.compatible='11.2.0.4.0'
*.control_files='/oracle/app/oracle/oradata/skystand/control01.ctl','/oracle/app/oracle/oradata/skystand/control02.ctl'
*.db_block_size=8192
*.db_domain=""
*.db_lost_write_protect='NONE'
*.db_name='sky'
*.diagnostic_dest='/oracle/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=skyXDB)'
*.open_cursors=300
*.pga_aggregate_target=104857600
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=419430400
*.undo_tablespace='UNDOTBS1'
```

```
DB_UNIQUE_NAME=skystand
LOG_ARCHIVE_CONFIG='DG_CONFIG=(skyprim,skystand)'
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/app/oracle/oradata/archive/skystand/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=skystand'
LOG_ARCHIVE_DEST_STATE_1=ENABLE
LOG_ARCHIVE_DEST_2='SERVICE=skyprim ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=skyprim'
LOG_ARCHIVE_DEST_STATE_2=ENABLE
FAL_SERVER=skyprim
DB_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky/', '/oracle/app/oracle/oradata/skystand/'
LOG_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky/', '/oracle/app/oracle/oradata/skystand/'
STANDBY_FILE_MANAGEMENT=AUTO
```

3. 1. 9 创建 Physical Standby Database 所需目录 (Standby Database)

```
[oracle@rac11g2 ~]$ cd $ORACLE_BASE/admin
[oracle@rac11g2 admin]$ mkdir -p skystand/adump/

[oracle@rac11g2 admin]$ cd $ORACLE_BASE/oradata
[oracle@rac11g2 oradata]$ mkdir -p skystand
[oracle@rac11g2 oradata]$ mkdir -p ./archive/skystand/
```

3. 1. 10 启动 Physical Standby Database (Standby Database)

```
[oracle@rac11g2 ~]$ export ORACLE_SID=skystand
[oracle@rac11g2 ~]$ sqlplus / as sysdba
SQL> startup nomount pfile=/home/oracle/sky.pfile
```

3. 1. 11 使用 RMAN 执行 DUPLICATE (Primary Database)

```
[oracle@rac11g1 ~]$ vi create_standby.rman
```

```
run
{
```

```
allocate channel prmy1 type disk;  
allocate channel prmy2 type disk;  
allocate channel prmy3 type disk;  
allocate channel prmy4 type disk;  
allocate auxiliary channel stby type disk;  
duplicate target database for standby from active database;  
}
```

```
[oracle@rac11g1 ~]$ rman  
RMAN> connect target sys/oracle  
RMAN> connect auxiliary sys/oracle@skystand  
RMAN> @create_standby.rman
```

RMAN DUPLICATE 的一些说明:

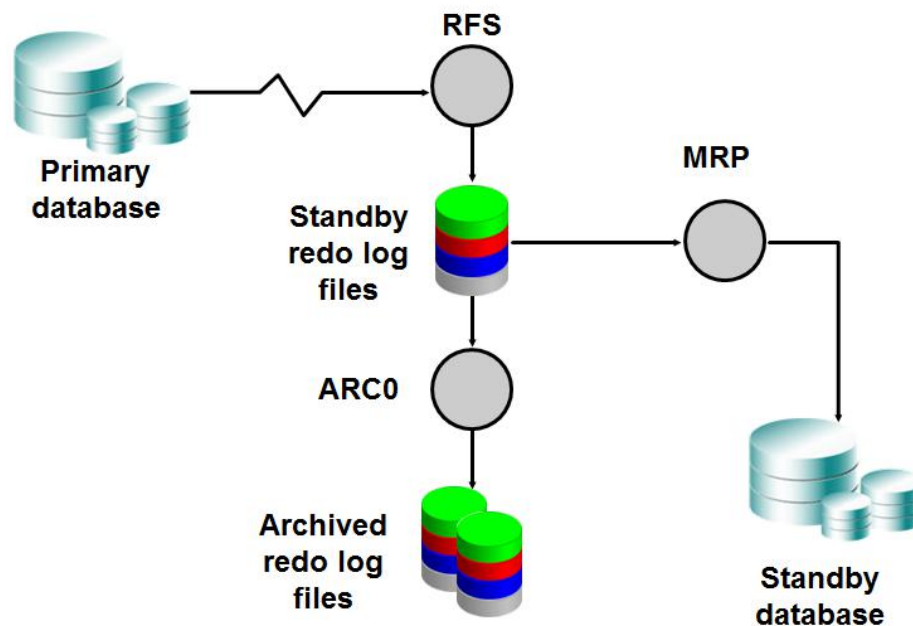
自动拷贝口令文件。

Data Guard 环境中的数据库，通过 DB_UNIQUE_NAME 进行唯一区分。

RMAN 不自动进行 standby database 的恢复操作，需要指定 DORECOVER 选项。

```
DUPLICATE TARGET DATABASE  
FOR STANDBY  
FROM ACTIVE DATABASE  
DORECOVER  
SPFILE  
SET "db_unique_name"="foou" COMMENT "Is a duplicate"  
SET LOG_ARCHIVE_DEST_2="service=inst3 ASYNC REGISTER  
VALID_FOR=(online_logfile,primary_role)"  
SET FAL_SERVER="inst1" COMMENT "Is primary"  
NOFILENAMECHECK;
```

3.1.12 启用 Real-Time Apply (Standby Database)



```
SQL> ALTER DATABASE OPEN;
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

3.1.13 防止 Primary Database 数据库冲突 (Primary/Standby Database)

Set the DB_LOST_WRITE_PROTECT initialization parameter on the primary and standby databases to enable the database server to record buffer cache block reads in the redo log so that lost writes can be detected.

```
SQL> ALTER SYSTEM SET DB_LOST_WRITE_PROTECT=TYPICAL;
```

3.2 物理备用数据库监控

3.2.1 查看备用数据库日志接收情况

备用数据库:

```
SQL> SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME, APPLIED FROM  
V$ARCHIVED_LOG ORDER BY SEQUENCE#;
```

主数据库:

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

备用数据库:

```
SQL> SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME, APPLIED FROM  
V$ARCHIVED_LOG ORDER BY SEQUENCE#;
```

3.3 物理备用数据库管理

3.3.1 物理备用数据库启停

➤ 启动

SQL*Plus STARTUP 命令启动物理备用数据库，不指定任何参数的情况下，启动模式为 read-only。

物理备用数据库可以在 MOUNT 和 OPEN 状态下接收主数据库日志。

如果 Redo Apply 启用以后无法从主数据库接收到日志数据，抛出 ORA-01112 错误。这表明备用数据库无法确定开始的日志序列，这种情况需要进行手工日志注册。

➤ 停止

如果主数据库处于运行状态，首先将主数据库的备用数据库归档目录设置为延迟，并进行一次日志切换，再停止备用数据库。

3.3.2 打开物理备用数据库

Active Data Guard 需要额外的授权。

3.3.2.1 Real-time 查询

启用 Oracle Active Data Guard 的 Real-time 查询特性,需要数据库的 COMPATIBLE 参数 ≥ 11.0 。

如果数据库在 MOUNT 状态下进行日志应用，物理备用数据库无法打开。需要先取消日志恢复。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```



```
SQL> ALTER DATABASE OPEN;

SQL> SELECT OPEN_MODE FROM V$DATABASE;

OPEN_MODE
-----
READ ONLY

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT;

SQL> SELECT OPEN_MODE FROM V$DATABASE;

OPEN_MODE
-----
READ ONLY WITH APPLY
```

3. 3. 2. 2 监控 Real-time 查询环境下的日志延迟

```
SQL> SELECT NAME, VALUE, DATUM_TIME, TIME_COMPUTED
       FROM V$DATAGUARD_STATS WHERE NAME LIKE 'apply lag';
```

NAME	VALUE	DATUM_TIME	TIME_COMPUTED
apply lag	+00 00:00:00	08/21/2015 03:23:17	08/21/2015 03:23:18

apply lag 度量用于计算从主数据库接收数据的周期性。**DATUM_TIME** 字段表示最近一次备用数据库接收到的日志。**TIME_COMPUTED** 记录 **apply lag** 计算的时间。这两个时间的间隔不应该大于 30 秒。

```
SQL> SELECT * FROM V$STANDBY_EVENT_HISTOGRAM
       WHERE NAME = 'apply lag' AND COUNT > 0;
```

NAME	TIME UNIT	COUNT	LAST_TIME_UPDATED
apply lag	0 seconds	674	08/21/2015 03:34:11
apply lag	23 minutes	2	08/21/2015 03:22:54

3.3.2.3 配置 Real-time 查询应用延迟的偏差

STANDBY_MAX_DATA_DELAY 是 11gr2 中对 Active Data Guard 的最大增强(buffer)之一, 这是一个可以在会话级别指定的参数, 该参数指定了在 Primary Database 已 commit 提交的变化与 standby Database 数据库上涉及相关变化的查询之间所允许的时间延迟, 单位为 second 秒。

注意事项:

1. 该参数无法为 SYS 用户所用, 在 SYS 用户的 SESSION 下设置该参数将被忽略
2. 若没有指定 STANDBY_MAX_DATA_DELAY, 即使用其默认值 NONE, 那么无论主备用数据库之间有多大的延迟, 在 Physical Standby 上的查询都会被执行
3. 若查询延迟超过 STANDBY_MAX_DATA_DELAY 所指定的值那么, 将报 ORA-03172 错误

配置 Primary 与 Standby 数据库之间的实时查询或者说零延迟查询有以下注意事项:

1. 只有特定的应用程序才会对数据延迟有零容忍的需求, 注意你的应用程序是否有如此苛刻的要求
2. 在 Standby 数据库上执行的查询语句必须返回和主数据库上查询的完全一致的结果
3. 必须设置 STANDBY_MAX_DATA_DELAY 为 0
4. 在查询开始的那一刻, Standby 数据库必须同步到与 Primary 数据库一致的 Current Scn
5. 若结果没有在 200ms 内返回, 则查询会因 ORA-03172 而终止
6. Primary 数据库必须采用最大可用(max availability)或最大保护(maximum protection)模式
7. redo 传输必须使用 SYNC 选项
8. 必须启用 Real-Time Query 特性

```
SQL> ALTER SESSION SET STANDBY_MAX_DATA_DELAY=2;
```

通过登录触发器设置该会话属性:

```
CREATE OR REPLACE TRIGGER AUTO_SMDD
AFTER LOGON ON USER.SCHEMA
BEGIN
  IF (SYS_CONTEXT('USERENV', 'DATABASE_ROLE') IN ('PHYSICAL STANDBY'))
  THEN
    EXECUTE IMMEDIATE 'alter session set standby_max_data_delay=5';
```

```
END IF;  
END;  
/
```

注意以上 trigger 只需要在 Primary Database 上以应用相关用户身份建立即可，会同步到 Standby 上。

3.3.2.4 Real-time 备用数据库强制日志同步

```
SQL> ALTER SESSION SYNC WITH PRIMARY;
```

3.3.3 需要手工维护的主数据库变更

3.3.3.1 表空间、数据文件添加

STANDBY_FILE_MANAGEMENT 为 AUTO 的情况下，表空间创建自动维护，。

```
MRP0: Background Media Recovery terminated with error 1274  
Fri Apr 8 10:00:22 2005  
Errors in file /u01/MILLER/MTS/dump/mts_mrp0_21851.trc:  
ORA-01274: cannot add datafile '/dev/raw/raw101' - file could not be created  
ORA-01119: error in creating database file '/dev/raw/raw101'  
ORA-27041: unable to open file  
Linux Error: 13: Permission denied  
Additional information: 1
```

```
SQL> SELECT NAME FROM V$DATAFILE;  
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=MANUAL;  
SQL> ALTER DATABASE CREATE DATAFILE  
'/u01/app/oracle/product/10.1.0/dbs/UNNAMED00007' AS '/dev/raw/raw101';  
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;  
SQL> RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

3.3.3.2 数据文件重命名

主数据库:

```
SQL> ALTER TABLESPACE tbs_4 OFFLINE;  
% mv /disk1/oracle/oradata/payroll/tbs_4.dbf /disk1/oracle/oradata/payroll/tbs_x.dbf
```

```
SQL> ALTER TABLESPACE tbs_4 RENAME DATAFILE  
'/disk1/oracle/oradata/payroll/tbs_4.dbf' TO '/disk1/oracle/oradata/payroll/tbs_x.dbf';  
SQL> ALTER TABLESPACE tbs_4 ONLINE;
```

备用数据库:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
SQL> SHUTDOWN;  
% mv /disk1/oracle/oradata/payroll/tbs_4.dbf /disk1/oracle/oradata/payroll/tbs_x.dbf  
SQL> STARTUP MOUNT;  
SQL> ALTER DATABASE RENAME FILE '/disk1/oracle/oradata/payroll/tbs_4.dbf' TO  
'/disk1/oracle/oradata/payroll/tbs_x.dbf';  
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

3.3.3.3 备用数据库添加、删除日志文件

1. Stop Redo Apply.
2. If the STANDBY_FILE_MANAGEMENT initialization parameter is set to AUTO, change the value to MANUAL.
3. Add or drop a log file group.

状态为 ACTIVE 的日志，需要进行手工清空

```
ALTER DATABASE CLEAR LOGFILE GROUP 3;
```

状态为 CURRENT 和 CLEARING_CURRENT 的日志，只有在角色切换后才可以进行删除。

4. Restore the STANDBY_FILE_MANAGEMENT initialization parameter and the Redo Apply options to their original states.
5. Restart Redo Apply.

3.4 转换为快照备用数据库

快照备用数据库从主数据库接收并归档重做数据，但不应用这些重做数据。当快照备用数据库重新转换为物理备用数据库后，会应用从主数据库接受到的日志，并且丢失快速备用数据库上的所有更改。

3.4.1 停止日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

3.4.2 确保数据库处于 MOUNT 状态，而不是 OPEN

3.4.3 确保配置了闪回区

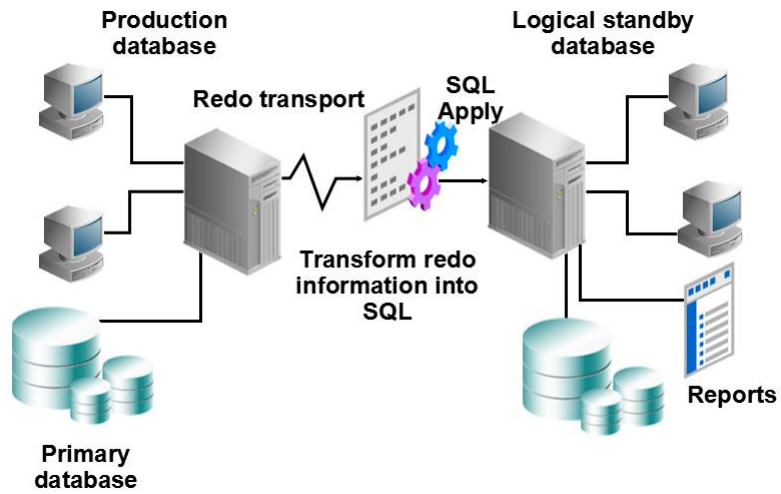
3.4.4 执行转换操作

```
SQL> ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;
```

3.4.5 执行转换回退操作

```
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
```

4 Logical Standby Database



5 Data Guard 保护模式

5.1 保护模式类型

5.1.1 Maximum Availability

最高可用性在保持主数据库可用的同时，提供级别稍低的数据保护。与最大保护一样，重做数据通过 SYNC 重做传输服务同步传输到备用数据库。只有当 Data Guard 确认事务数据在至少一台备用服务器的磁盘上可用之后，日志写入器才会确认主数据库已提交事务。但与最大保护模式不同的是，如果最后参与的备用数据库不可用（例如出现网络连接问题或备用故障），主数据库处理将继续进行。断开连接时，与主数据库相比，备用数据库可能暂时滞后。而当再次建立连接时，Data Guard 将自动重新同步备用数据库和主数据库。

由于同步的重做传输，这种保护模式可能影响响应时间和吞吐量。通过配置一个带宽足够应付高峰事务负载的低延迟网络，可以将这种影响降至最小。

最高可用性模式适用于想要确保获得零数据丢失保护但不想让生产数据库受网络或备用服务器故障影响的企业。如果在最初的网络或备用故障得到解决和配置重新同步之前，又一个故障随后影响了生产数据库，则会发生数据丢失，采用这种保护模式的企业要接受这样的事实。

5.1.2 Maximum Performance

最高性能是默认的保护模式。

与最高可用性模式相比，它提供了稍差的数据保护，但可为主数据库提供更高的性能。在这种模式下，当主数据库处理事务时，重做数据通过 ASYNC 重做传输异步传输到备用数据库。主数据库上的日志写入器不用等待备用数据库确认即可确认到客户端应用程序的提交。这消除了网络往返行程和备用磁盘输入/输出可能生成的主数据库上的响应时间开销。如果任何备用目标不可用，处理将在主数据库上继续进行，对性能只有很小的影响或没有影响。

在正常操作中，可能丢失的数据量只限于在主数据库和备用数据库之间传输的数据量——这个量由处理主数据库生成的重做数据量的网络容量决定。如果有足够的带宽，可能丢失的总数据量非常小或为零。

当主数据库上的可用性和性能比丢失少量数据的风险更重要时，应该使用最高性能模式。这种模式还适合于客户网络固有延迟高到可能限制同步重做传输的适用性时的 Data Guard 部署。

5.1.3 Maximum Protection

最大保护提供最高级别的数据保护。最大保护使用 SYNC 重做传输将重做记录同步传输到备用数据库。只有当 Data Guard 确认事务数据安全保存到至少一台备用服务器的磁盘上之后，主数据库上的日志写入器进程才会确认到客户端应用程序的提交。

由于重做传输的同步特性，最大保护模式可能影响主数据库的响应时间。通过配置一个带宽足够应付高峰事务负载的低延迟网络，可以将这种影响降至最小。例如，金融机构和彩票系统就部署了这种最高级别的数据保护模式。

强烈建议，最大保护模式应至少配置两个备用数据库。这样，如果一个备用目标出现故障，另外一个可用的目标仍可向主数据库做出确认，而不中断生产。考虑使用这种保护模式的用户必须接受的事实是，如果至少一个备用数据库无法返回重做已被接收并写入磁盘的确认，主数据库将停滞（并最终崩溃）。最大保护的规则规定这一行为的目的是在有多个故障的情况下（例如，主数据库和备用数据库之间的网络出现故障，然后又有一个事件导致主站点发生故障）实现零数据丢失。如果您无法接受这一要求，并希望在得到零数据丢失保护的同时，主数据库即使在备用数据库无法确认数据受到保护的情况下也能保持可用，可以使用接下来讨论的最高可用性保护模式。

5.2 修改保护模式

5.2.1 确认 REDO 传输参数

至少有一个 Standby Database 的日志传输满足数据保护模式的需求。

Maximum Availability	Maximum Performance	Maximum Protection
AFFIRM	NOAFFIRM	AFFIRM
SYNC	ASync	SYNC
DB_UNIQUE_NAME	DB_UNIQUE_NAME	DB_UNIQUE_NAME

主数据库：

```
SQL> ALTER SYSTEM SET
```

```
LOG_ARCHIVE_DEST_2='SERVICE=skystand AFFIRM SYNC
```

```
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=skystand';
```

备用数据库：


```
SQL> ALTER SYSTEM SET
```

```
LOG_ARCHIVE_DEST_2='SERVICE=skyprim AFFIRM SYNC
```

```
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=skyprim';
```

5.2.2 修改保护模式

该命令在主数据库执行：

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE {PROTECTION |  
AVAILABILITY | PERFORMANCE};
```

5.2.3 查看数据库保护模式

```
SQL> SELECT PROTECTION_MODE FROM V$DATABASE;
```

```
PROTECTION_MODE
```

```
-----
```

```
MAXIMUM AVAILABILITY
```

6 Redo 传输服务

Redo transport services 负责在数据库之间自动传输 REDO 数据。REDO 传输的目标支持：

- Oracle Data Guard standby databases
- Archive Log repository
- Oracle Streams downstream capture databases
- Oracle Change Data Capture staging databases

REDO 传输的模式：

- Synchronous

同步重做传输 (SYNC) 要求主数据库上的日志写入器等待 LNS 确认备用数据库已经接收重做数据并已将其写入备用重做日志，然后才能确认到客户端应用程序的提交。这确保了提交的所有事务都在磁盘上，并在备用位置受到保护。

该模式在 Maximum Protection 和 Maximum Availability 的情况下使用。

- Asynchronous

异步重做传输 (ASYNC) 不要求主数据库上的日志写入器等待备用数据库确认重做已被写入磁盘。确认到客户端应用程序的提交与重做传输是异步进行的。

该模式在 Maximum Performance 的情况下使用。

6.1 Redo 传输的安全

Redo 传输服务，使用 SSH 或者远程口令文件进行会话安全认证。在 Data Guard 环境中，standby database 数据库使用 primary database 的口令文件。

当添加 SYSOPER 或者 SYSDBA 权限的用户后，需要重新同步口令文件。口令修改后，也需要重新同步该文件。所有日志传输的数据库口令文件要保持一致。默认情况下，使用 SYS 用户进行验证。

REDO_TRANSPORT_USER 参数可以控制 ORACLE 使用拥有 SYSOPER 角色的用户进行日志传输。

6.2 发送 Redo Data

LOG_ARCHIVE_DEST_n 指定归档的位置。

LOG_ARCHIVE_DEST_STATE_n 指定对应的归档位置是否可用。

参数详细配置，参考 DG 核心参数部分内容，见 [2.3.3](#)

```
LOG_ARCHIVE_DEST_2='SERVICE=CHICAGO ASYNC NOAFFIRM  
VALID_FOR=(ONLINE_LOGFILE, PRIMARY_ROLE) REOPEN=60  
COMPRESSION=ENABLE DB_UNIQUE_NAME=CHICAGO'
```

监控视图：

```
SELECT * FROM V$ARCHIVE_DEST WHERE DESTINATION IS NOT NULL;
```

6.3 接收 Redo Data

Standby redo log 用来接收从其他数据库传过来的 redo 数据。RFS 进程将其他数据库发送过来的 redo 数据，写入当前的 standby redo log。Standby redo log 的数量要大于源库的 redo log 数量。大小要一致。建议主备数据库都配置 standby redo log。

```
SQL> SELECT GROUP#, BYTES FROM V$LOG;
```

```
SQL> SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

通过下面的命令添加 standby redo log:

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE ('/oracle/dbs/slog1.rdo') SIZE 500M;
```

RAC 环境下，需要指定实例：

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 SIZE 500M;
```

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 SIZE 500M;
```

6.4 级联 Redo 传输

级联的限制：

只有物理备用数据库可以进行 redo 级联；

Data Guard broker 不被级联目标支持;

Primary Database

```
DB_UNIQUE_NAME=boston
FAL_SERVER=boston2
LOG_ARCHIVE_CONFIG='DG_CONFIG=(boston,boston2,denver)'
LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston'
LOG_ARCHIVE_DEST_2='SERVICE=boston2 SYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=boston2'
```

Cascading Physical Standby Database

```
DB_UNIQUE_NAME=boston2
FAL_SERVER=boston
LOG_ARCHIVE_CONFIG= 'DG_CONFIG=(boston,boston2,denver)'
LOG_ARCHIVE_DEST_1='LOCATION= USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston2'
LOG_ARCHIVE_DEST_2= 'SERVICE=denver
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=denver'
```

Cascaded Physical Standby Database

```
DB_UNIQUE_NAME=denver
FAL_SERVER=boston2
LOG_ARCHIVE_CONFIG='DG_CONFIG=(boston,boston2,denver)'
LOG_ARCHIVE_DEST_1='LOCATION= USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=denver'
```

6.5 监控 Redo 传输服务

6.5.1 查看当前最大归档日志

```
SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG GROUP BY
THREAD#;
```

6.5.2 查看各日志传输目标状态

```
SELECT DESTINATION, STATUS, ARCHIVED_THREAD#, ARCHIVED_SEQ#
FROM V$ARCHIVE_DEST_STATUS
WHERE STATUS <> 'DEFERRED' AND STATUS <> 'INACTIVE';
```

6.5.3 查询归档目标中不一致的日志

```
SELECT LOCAL.THREAD#, LOCAL.SEQUENCE#
FROM (SELECT THREAD#, SEQUENCE# FROM V$ARCHIVED_LOG WHERE DEST_ID = 1) LOCAL
WHERE LOCAL.SEQUENCE# NOT IN (SELECT SEQUENCE#
                                FROM V$ARCHIVED_LOG
                                WHERE DEST_ID = 2
                                AND THREAD# = LOCAL.THREAD#);
```

6.5.4 跟踪日志传输

Level	Meaning
0	Disables archived redo log tracing (default setting)
1	Tracks archiving of log files
2	Tracks archive status by archive log file destination
4	Tracks archive operational phase
8	Tracks archive log destination activity
16	Tracks detailed archive log destination activity
32	Tracks archive log destination parameter modifications
64	Tracks ARCn process state activity
128	Tracks FAL server process activity
256	Tracks RFS Logical Client

512	Tracks LGWR redo shipping network activity
1024	Tracks RFS physical client
2048	Tracks RFS/ARCn ping heartbeat
4096	Tracks real-time apply activity
8192	Tracks Redo Apply activity (media recovery or physical standby)
16384	Tracks archive I/O buffers
32768	Tracks LogMiner dictionary archiving

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_TRACE=15;
```

6.5.5 查看同步 Redo 传输响应时间

```
SQL> SELECT FREQUENCY, DURATION FROM
V$REDO_DEST_RESP_HISTOGRAM WHERE DEST_ID=2 AND FREQUENCY > 1;
SQL> SELECT MAX(DURATION) FROM V$REDO_DEST_RESP_HISTOGRAM
WHERE DEST_ID = 2 AND FREQUENCY > 1;
SQL> SELECT MIN(DURATION) FROM V$REDO_DEST_RESP_HISTOGRAM WHERE
DEST_ID= 2 AND FREQUENCY > 1;
```

6.6 Redo Gap 检测和处理

当 redo 传输中断时，会产生 Redo Gap，redo 传输服务会自动检测 Redo Gap 并且将 redo 重新发送到目标。

Redo 传输服务通过下面的技术减少 Redo Gap 传输的时间：

- Redo Transport Compression
LOG_ARCHIVE_DEST_n 的 COMPRESSION 参数。
- Parallel Redo Transport Network Sessions
LOG_ARCHIVE_DEST_n 的 MAX_CONNECTIONS 参数。

6.6.1 手工处理物理备用数据库 Redo Gap

1. 检查物理备用数据库的 Redo Gap

```
SQL> SELECT * FROM V$ARCHIVE_GAP;
```

THREAD#	LOW_SEQUENCE#	HIGH_SEQUENCE#
1	7	10

2. 检查主数据库对应的归档信息

```
SQL> SELECT NAME FROM V$ARCHIVED_LOG WHERE THREAD# = 1 AND  
DEST_ID = 1 AND SEQUENCE# BETWEEN 7 AND 10;
```

NAME
/primary/thread1_dest/archr_1_7.arc
/primary/thread1_dest/archr_1_8.arc
/primary/thread1_dest/archr_1_9.arc

3. 拷贝归档文件到备用数据库并注册

```
SQL> ALTER DATABASE REGISTER LOGFILE '/physical_standby1/thread1_dest/archr_1_7.arc';  
SQL> ALTER DATABASE REGISTER LOGFILE '/physical_standby1/thread1_dest/archr_1_8.arc';  
SQL> ALTER DATABASE REGISTER LOGFILE '/physical_standby1/thread1_dest/archr_1_9.arc';
```

6.6.2 手工处理逻辑备用数据库 Redo Gap

1. 查询逻辑备用数据库 Redo Gap

```
SELECT THREAD#, SEQUENCE#, FILE_NAME  
FROM DBA_LOGSTDBY_LOG L  
WHERE NEXT_CHANGE# NOT IN  
(SELECT FIRST_CHANGE# FROM DBA_LOGSTDBY_LOG WHERE L.THREAD# = THREAD#)  
ORDER BY THREAD#, SEQUENCE#;
```

THREAD#	SEQUENCE#	FILE_NAME
1	6	/disk1/oracle/dbs/log-1292880008_6.arc
1	10	/disk1/oracle/dbs/log-1292880008_10.arc

2. 手工注册日志

```
SQL> ALTER DATABASE REGISTER LOGICAL LOGFILE '/disk1/oracle/dbs/log-1292880008_7.arc';  
SQL> ALTER DATABASE REGISTER LOGICAL LOGFILE '/disk1/oracle/dbs/log-1292880008_8.arc';  
SQL> ALTER DATABASE REGISTER LOGICAL LOGFILE '/disk1/oracle/dbs/log-1292880008_9.arc';
```

6.7 Redo 传输等待事件

Wait Event	Description
LNS wait on ATTACH	Total time spent waiting for redo transport sessions to be established to all ASYNC and SYNC redo transport destinations
LNS wait on SENDREQ	Total time spent waiting for redo data to be written to all ASYNC and SYNC redo transport destinations
LNS wait on DETACH	Total time spent waiting for redo transport connections to be terminated to all ASYNC and SYNC redo transport destinations

7 Apply Services

重做数据被传送到备用站点后，**Data Guard** 应用服务提供两种不同的方法用于将数据应用于备用数据库：**Data Guard** 重做应用（物理备用）或 **Data Guard SQL** 应用（逻辑备用）。**Apply services** 自动应用从主数据库接收到日志到备用数据库。

7.1 Apply Services 类型

➤ **Redo Apply** (重做应用，用于物理备用数据库)

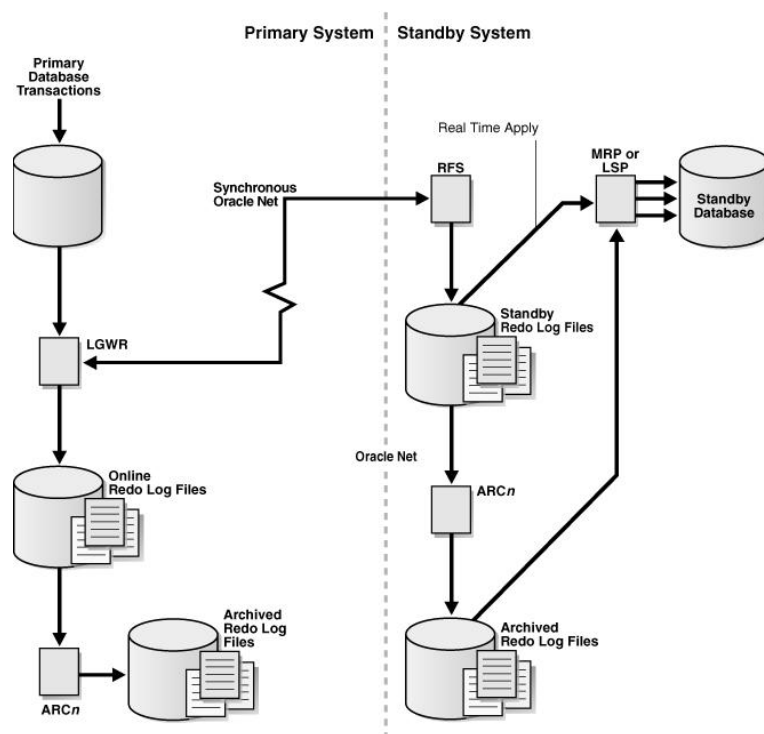
物理备用数据库具有磁盘上的数据库结构，该结构每块都与主数据库完全一样，使用 **Oracle** 介质恢复进行更新。

➤ **SQL Apply** (SQL 应用，用于逻辑备用数据库)

逻辑备用数据库是一个独立的数据库，它与主数据库包含相同的数据，使用 **SQL** 语句进行更新。每种方法都有其不同的优势，用户可根据自己的需求灵活选择。本文后面部分将详细讨论其差异。

7.2 Apply Services 配置

7.2.1 Real-Time Apply



Real-time apply 要求 standby database 配置 standby database log 并且处于归档模式。

➤ physical standby databases

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE;
```

➤ logical standby databases

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

7.2.2 Time Delay Apply

如果 Real-Time Apply 被启用, Time Delay 将被忽略。通过设置主数据库和备用数据库 LOG_ARCHIVE_DEST_n 初始化参数的 DELAY=minutes 属性来实现归档日志文件的延迟应用。DELAY 默认时间为 30 分钟。

取消延迟配置:

➤ physical standby databases

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE NODELAY;
```

➤ **logical standby databases**

```
ALTER DATABASE START LOGICAL STANDBY APPLY NODELAY;
```

7.3 物理备用数据库的日志应用

默认情况下，应用 archived redo log 中的日志数据。当启用 Real-Time Apply 时，可以直接应用由 RFS 写入的 standby redo log。

7.3.1 启动日志应用

执行日志应用之前，确保备用数据库处于 MOUNTED 状态。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;
```

在后台进行日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
DISCONNECT;
```

启动 Real-Time Apply

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

7.3.2 停止日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

7.4 逻辑备用数据库的日志应用

7.4.1 启动日志应用

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

7.4.2 停止日志应用

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

8 角色转换

查看当前数据库角色

```
SQL> SELECT NAME, OPEN_MODE, DATABASE_ROLE FROM V$DATABASE;
```

➤ 主数据库

NAME	OPEN_MODE	DATABASE_ROLE
SKY	READ WRITE	PRIMARY

➤ 备用数据库

NAME	OPEN_MODE	DATABASE_ROLE
SKY	READ ONLY WITH APPLY PHYSICAL STANDBY	

8.1 物理备用数据库 Switchover

转换用于减少计划内中断期间（例如，操作系统或硬件升级，或者滚动升级 Oracle 数据库软件和补丁集）的主数据库停机时间。

转换操作要求断开所有用户会话与主数据库的连接。之后，将主数据库转换为备用角色，然后再将备用数据库转换为主数据库角色。

8.1.1 切换前的准备

1. 参数确认

确保 LOG_ARCHIVE_DEST_n/LOG_ARCHIVE_DEST_STATE_n 参数正确配置

2. 日志应用确认

在主数据库上检查日志传输状态为正常，没有日志间隔。

```
SQL> SELECT STATUS, GAP_STATUS FROM V$ARCHIVE_DEST_STATUS WHERE DEST_ID = 2;
```

STATUS	GAP_STATUS
VALID	NO GAP

- 3. RAC 环境下，要转换为备用数据库的主数据库，只保留一个实例。
- 4. Real-Time 模式下，备用数据库尽量使用 MOUNT 状态打开，加速日志的应用。

8.1.2 选择角色转换的备用数据库

查询备用数据库接收并应用主数据库日志的信息。

```
SQL> SET LINESIZE 120;
SQL> COLUMN NAME FORMAT A24;
SQL> COLUMN VALUE FORMAT A16;
SQL> COLUMN DATUM_TIME FORMAT A24;
SQL> SELECT NAME, VALUE, DATUM_TIME FROM V$DATAGUARD_STATS;
```

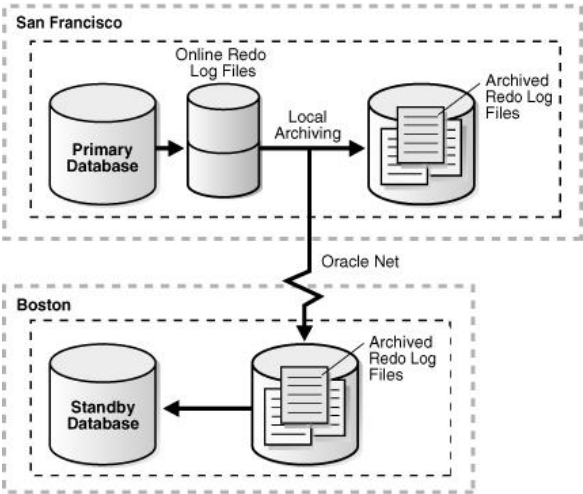
NAME	VALUE	DATUM_TIME
transport lag	+00 00:00:00	08/20/2015 07:52:05
apply lag	+00 00:00:00	08/20/2015 07:52:05
apply finish time		
estimated startup time	20	

DATUM_TIME 值不变，可能是主数据库和备用数据库之间的通讯出现故障。

8.1.3 执行切换

切换的过程分为两步：

8.1.3.1 切换之前的状态



备用数据库应用日志:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE;
```

主数据库状态:

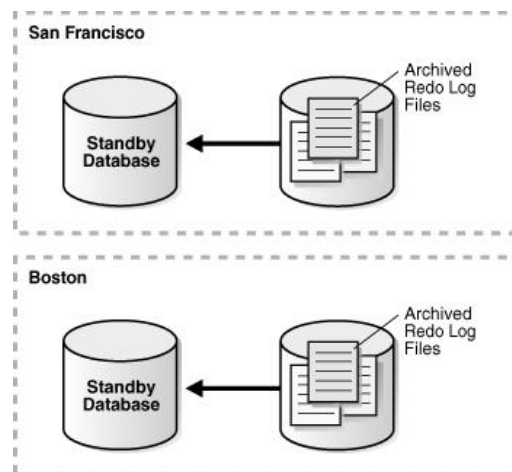
```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

```
SWITCHOVER_STATUS
```

```
SESSIONS ACTIVE
```

TO STANDBY 或 SESSIONS ACTIVE 状态，表明主数据库可以转换为备用数据库。

8.1.3.2 原主数据库切换为备用数据库



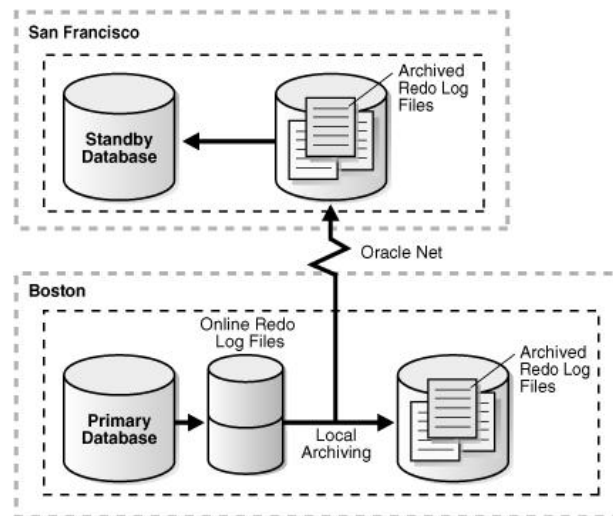
切换为备用数据库:

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY WITH  
SESSION SHUTDOWN;
```

启动新备用数据库到 MOUNT 状态

```
SQL> STARTUP MOUNT;
```

8.1.3.3 原备用数据库转换为主数据库



```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

```
SWITCHOVER_STATUS
```

```
SESSIONS ACTIVE
```

TO STANDBY 或 SESSIONS ACTIVE 状态，表明备用数据库可以转换为主数据库。

切换备用数据库到主数据库：

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WITH SESSION SHUTDOWN;
```

打开新主数据库：

```
SQL> ALTER DATABASE OPEN;
```

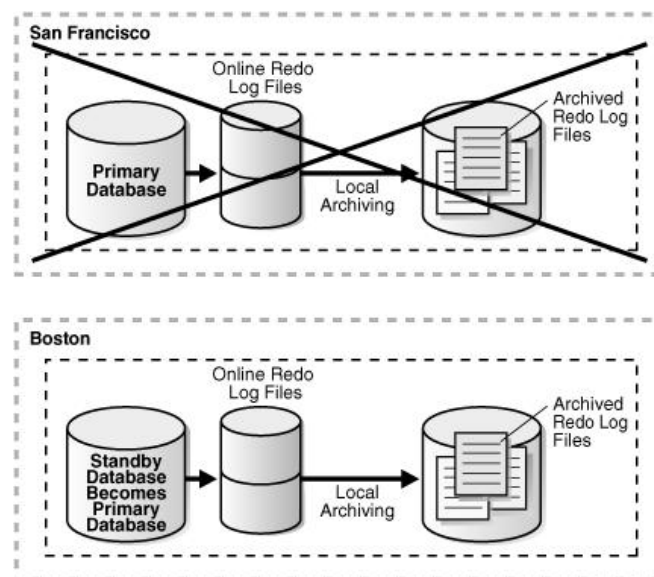
新备用数据库实时应用日志：

```
SQL> ALTER DATABASE OPEN;
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT FROM SESSION;
```

8.2 物理备用数据库 Failover

故障切换是在主数据库上出现计划外的故障时，将一个联机的备用数据库用作新的主数据库的操作。这种方法通过快速将备用数据库提升为主数据库角色来实现恢复时间目标，对影响主数据库的事件进行诊断和解决的时候不会出现停机。故障切换操作不需要重启备用数据库。此外，只要原来的主数据库上的数据库文件未被更改且数据库可重启，则可以使用闪回数据库恢复原来的主数据库，并将其重新同步为新的主数据库的一个备用数据库 — 不需要从备份恢复。



最大保护模式下的备用数据库，在进行故障切换之前，需要将备用数据库模式修改为最大性能：

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
```

8.2.1 执行故障切换

1. 刷新主数据库未发送到备用数据库的日志

主数据库可以在 MOUNT 的情况下，进行日志的刷新。目标数据库的名称为备用数据库的 DB_UNIQUE_NAME。

```
SQL> ALTER SYSTEM FLUSH REDO TO target_db_name;
```

2. 确认备用数据库最后的日志序号

```
SQL> SELECT UNIQUE THREAD# AS THREAD, MAX(SEQUENCE#) OVER  
(PARTITION BY THREAD#) AS LAST from V$ARCHIVED_LOG;
```


注册缺失的日志

```
SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';
```

3. 确认日志的 Gap

```
SQL> SELECT THREAD#, LOW_SEQUENCE#, HIGH_SEQUENCE# FROM  
V$ARCHIVE_GAP;
```

注册缺失的日志

```
SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';
```

4. 备用数据库取消日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

5. 备用数据库完成所有的日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;
```

如果有报错并无法解决，执行故障切换(这种情况会丢失数据)

```
SQL> ALTER DATABASE ACTIVATE PHYSICAL STANDBY DATABASE;
```

6. 检查备用数据库是否准备好切换为主数据库

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

7. 备用数据库切换为主数据库

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WITH SESSION  
SHUTDOWN;
```

8. 打开新主数据库

```
SQL> ALTER DATABASE OPEN;
```

9. 启动中断的备用数据库恢复

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

重新初始化主库

Reinstating a Physical Standby Using Backups Instead of Flashback (文档 ID 416310.1)
Step by Step Guide on How To Reinstat Failed Primary Database into Physical Standby
(文档 ID 738642.1)

8.3 逻辑备用数据库 Switchover

1. 检查是否可以切换为主数据库

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

2. 当前主数据库准备切换

```
SQL> ALTER DATABASE PREPARE TO SWITCHOVER TO LOGICAL STANDBY;
```

3. 目标备用数据库准备切换

```
SQL> ALTER DATABASE PREPARE TO SWITCHOVER TO PRIMARY;
```

4. 确定当前主数据库可以转换为备用数据库

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

5. 主数据库转换为备用数据库角色

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO LOGICAL STANDBY;
```

6. 确定备用数据库可以转换为主数据库

```
SQL> SELECT SWITCHOVER_STATUS FROM V$DATABASE;
```

7. 备用数据库转换为主数据库

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY;
```

8. 开启新备用数据库的日志应用

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

8.4 逻辑备用数据库 Failover

1. 刷新主数据库未传输的日志

```
SQL> ALTER SYSTEM FLUSH REDO TO target_db_name;
```

2. 注册备用数据库确实确实的日志

```
SQL> ALTER DATABASE REGISTER LOGICAL LOGFILE  
'/disk1/oracle/dbs/log-%r_%s_%t.arc';
```

3. 启用远程归档目标

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE  
SCOPE=BOTH;
```

4. 激活新的主数据库

```
SQL> ALTER DATABASE ACTIVATE LOGICAL STANDBY DATABASE FINISH APPLY;
```

5. 恢复故障切换后的其他备用数据库

6. 备份新的主数据库

7. 还原故障的主数据库

8.5 角色转换触发器

DB_ROLE_CHANGE 事件在角色转换时触发。

8.6 闪回数据库

在数据切换之后，可以通过 FLASHBACK DATABASE 将数据库转换到角色切换之前。如果闪回主数据库，那么所有备用数据库都需要闪回。当闪回主数据库和备用数据库后，不需要了解之前的切换关系，ORACLE 会自动处理过去的切换。

必须在切换之前启用闪回数据库。

9 物理备库和快照备库管理

9.1 开启和关闭物理备库

9.1.1 开启物理备库

当我们并没有从主库上获取 redo 的时候，备库就开始 redo apply，此时会报 ORA-01112。这表明 redo apply 无法进行介质恢复。当此类问题出现后，我们需要手动同步 redo logs，等同步完成后再进行 redo apply。

9.1.2 关闭物理备库

9.2 开启备库

9.2.1 实时查询

Oracle ADG 中参数 COMPATIBLE 必须设置成 11.0 或者更高才可以使用 real-time query 新特征。

在 oracle mount 状态的时候 redo apply active 时，物理备库是不可以 open 的，我们可以使用下来语句进行数据库 read-only open 状态，并且重新 redo apply。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

```
SQL> ALTER DATABASE OPEN;
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT;
```

9.2.1.1 在 real-time query 环境下，监控 oracle apply 延迟

```
SQL> SELECT name, value, datum_time, time_computed FROM  
V$DATAGUARD_STATS WHERE name like 'apply lag';
```

NAME	VALUE	DATUM_TIME	TIME_COMPUTED
-----	-----	-----	-----
apply lag	+00 00:00:00	05/27/2009 08:54:16	05/27/2009 08:54:17

apply lag 参数是备库一次从主库接受到数据的时间间隔, **DATUM_TIME** 参数是备库从主库上一次接受数据当时的时间戳, **TIME_COMPUTED** 参数是指 **apply lag** 的间隔估算。上面的这两个值的最大值是 30 秒, 如果大于 30 秒说明该值已经不准确了。

查询历史数据同步记录:

```
SQL> SELECT * FROM V$STANDBY_EVENT_HISTOGRAM WHERE NAME = 'apply lag' AND COUNT > 0;
```

9.2.1.2 配置 apply lag 的容忍度在 real-time query 环境中

会话层参数 **STANDBY_MAX_DATA_DELAY** 可以用于指定 **applylag** 的容忍度, 单位是秒。

10 DG 管理与监控

10.1 常用数据字典

10.1.1 V\$DATABASE

查询数据库的保护级别，角色等信息。

```
SQL> SELECT NAME, OPEN_MODE, DATABASE_ROLE FROM V$DATABASE;
```

主数据库:

NAME	OPEN_MODE	DATABASE_ROLE
SKY	READ WRITE	PRIMARY

备用数据库:

NAME	OPEN_MODE	DATABASE_ROLE
SKY	READ ONLY WITH APPLY PHYSICAL STANDBY	

```
SQL> SELECT PROTECTION_MODE, PROTECTION_LEVEL, DATABASE_ROLE  
ROLE, SWITCHOVER_STATUS FROM V$DATABASE;
```

主数据库:

PROTECTION_MODE	PROTECTION_LEVEL	ROLE	SWITCHOVER_STATUS
MAXIMUM AVAILABILITY	MAXIMUM AVAILABILITY	PRIMARY	SESSIONS ACTIVE

备用数据库:

PROTECTION_MODE	PROTECTION_LEVEL	ROLE	SWITCHOVER_STATUS
MAXIMUM AVAILABILITY	MAXIMUM AVAILABILITY	PHYSICAL STANDBY	NOT ALLOWED

快速切换属性:

```
SELECT FS_FAILOVER_STATUS          "FSFO STATUS",
```

```

        FS_FAILOVER_CURRENT_TARGET    TARGET,
        FS_FAILOVER_THRESHOLD          THRESHOLD,
        FS_FAILOVER_OBSERVER_PRESENT  "OBSERVER PRESENT"
FROM V$DATABASE;

```

10.1.2 V\$MANAGED_STANDBY

物理备用数据库的日志传输和日志应用状态

```
SQL> SELECT PROCESS, STATUS, THREAD#, SEQUENCE#, BLOCK#, BLOCKS
FROM V$MANAGED_STANDBY;
```

PROCESS	STATUS	THREAD#	SEQUENCE#	BLOCK#	BLOCKS
ARCH	CLOSING	1	203	6144	218
ARCH	CLOSING	1	205	6144	508
ARCH	CONNECTED	0	0	0	0
ARCH	CLOSING	1	204	6144	258
RFS	IDLE	0	0	0	0
RFS	IDLE	1	206	795	1
RFS	IDLE	0	0	0	0
MRP0	APPLYING_LOG	1	206	795	102400

10.1.3 V\$ARCHIVED_LOG

备用数据库上接收到的未被应用的日志文件

```
SQL> SELECT THREAD#, SEQUENCE#, FIRST_CHANGE#, NEXT_CHANGE# FROM
V$ARCHIVED_LOG WHERE APPLIED = 'NO';
```

10.1.4 V\$LOG_HISTORY

归档日志的历史信息

```
SQL> SELECT THREAD#, SEQUENCE#, FIRST_CHANGE#, NEXT_CHANGE#
FROM V$LOG_HISTORY;
```

10.1.5 V\$ARCHIVE_DEST

归档传输目的地信息

```
SQL> SELECT * FROM V$ARCHIVE_DEST WHERE STATUS='VALID';
```

通过 ERROR 字段，可以看到归档失败的原因。

10.1.6 V\$ARCHIVE_GAP

```
SQL> SELECT * FROM V$ARCHIVE_GAP;
```

10.1.7 V\$DATAGUARD_CONFIG

```
SQL> SELECT * FROM V$DATAGUARD_CONFIG;
```

10.1.8 V\$DATAGUARD_STATUS

Data Guard 事件信息，这些事件会写入告警日志或服务进程日志。

```
SQL> SELECT * FROM V$DATAGUARD_STATUS;
```

10.1.9 V\$DATAGUARD_STATS

```
SQL> SELECT * FROM V$DATAGUARD_STATS;
```

10.1.10 V\$STANDBY_LOG

备用日志信息

```
SQL> SELECT * FROM V$STANDBY_LOG;
```

10.1.11 V\$REDO_DEST_RESP_HISTOGRAM

REDO 日志传输的响应时间。

```
SELECT MAX(DURATION)
  FROM V$REDO_DEST_RESP_HISTOGRAM
 WHERE DEST_ID = 2
    AND FREQUENCY > 1;
```


10.2 Data Guard 启停

10.2.1 启动

- 启动备库

```
SQL> STARTUP;
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE PARALLEL 4 DISCONNECT FROM SESSION;
```

- 启动主库

```
SQL> STARTUP;
```

10.2.2 停止

- 主库进行日志切换

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

- 调整主库传输到备库的日志目标状态

```
SQL> ALTER SYSTEM SET log_archive_dest_state_2=DEFER;
```

- 停止备库

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

```
SQL> SHUTDOWN IMMEDIATE;
```

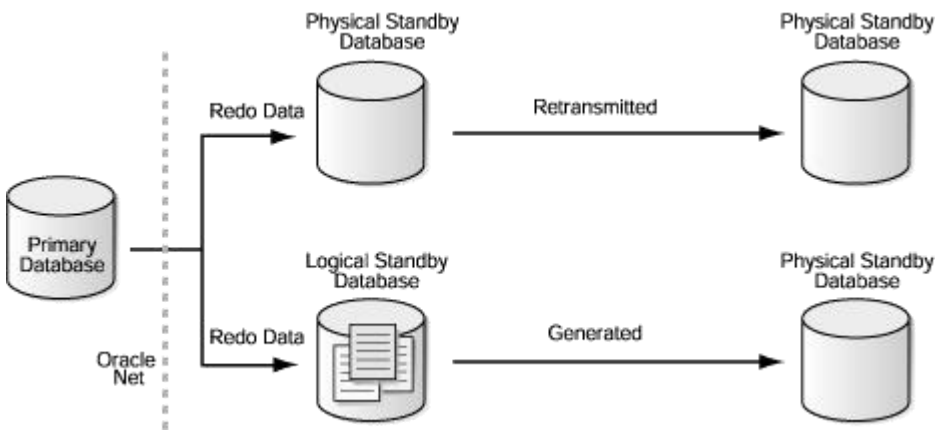
- 停止主库

```
SQL> SHUTDOWN IMMEDIATE;
```

11 DG 应用案例

11.1 RMAN 备份恢复

11.2 级联 DG



The following Data Guard configurations using cascaded destinations are supported.

- 1. Primary Database > Physical Standby Database with cascaded destination > Physical Standby Database
- 2. Primary Database > Physical Standby Database with cascaded destination > Logical Standby Database

11.2.1 环境信息规划

Database	HOST	DB_UNIQUE_NAME	Net Service Name	DB_NAME
主库	rac11g1	skyprim	skyprim	sky
物理备库	rac11g2	skystand	skystand	
级联备库	rac11g2	skycase	skycase	

资料库	rac11g1	catalog	catalog	catalog
-----	---------	---------	---------	---------

主库和 RMAN 恢复目录库位于同一台主机上，物理备库和级联备库都在一台主机上面。

11.2.2 Cascading Standby Database 配置要求

FAL_SERVER 参数设置为 Primary Database 或者直接从 Primary Database 接收日志的 Standby Database。

11.2.2.1 简单级联设置实例：

Database	DB_UNIQUE_NAME	Net Service Name	DB_NAME
主库	boston	boston	sky
物理备库	boston2	boston2	
级联备库	denver	denver	

➤ Primary Database

日志传输方向：主库 → 物理备库

DB_UNIQUE_NAME=boston

FAL_SERVER=boston2

LOG_ARCHIVE_CONFIG='DG_CONFIG=(boston,boston2,denver)'

LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST

VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston'

LOG_ARCHIVE_DEST_2='SERVICE=boston2 SYNC

VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=boston2'

➤ Cascading Physical Standby Database

日志传输方向：物理备库 → 级联备库

DB_UNIQUE_NAME=boston2

```
FAL_SERVER=boston
LOG_ARCHIVE_CONFIG= 'DG_CONFIG=(boston,boston2,denver)'

LOG_ARCHIVE_DEST_1='LOCATION= USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=boston2'
LOG_ARCHIVE_DEST_2= 'SERVICE=denver
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=denver'
```

➤ Cascaded Physical Standby Database

日志传输方向：级联备库不传递

```
DB_UNIQUE_NAME=denver

FAL_SERVER=boston2
LOG_ARCHIVE_CONFIG='DG_CONFIG=(boston,boston2,denver)'
LOG_ARCHIVE_DEST_1='LOCATION= USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=denver'
```

11.2.3 主库准备工作

11.2.3.1 开启数据库 FORCE LOGGING (Primary Database)

```
SQL> SHOW PARAMETER INSTANCE_NAME;
```

NAME	TYPE	VALUE
instance_name	string	sky

```
SQL> SELECT NAME, LOG_MODE, FORCE_LOGGING FROM V$DATABASE;
```

NAME	LOG_MODE	FOR
SKY	NOARCHIVELOG	YES

开启:

```
SQL> ALTER DATABASE FORCE LOGGING;
```

关闭:

```
SQL> ALTER DATABASE NO FORCE LOGGING;
```

11.2.3.2 启动归档模式 (Primary Database)

```
SQL> ALTER SYSTEM SET  
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/archive/sky/';
```

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP MOUNT;
```

```
SQL> ALTER DATABASE ARCHIVELOG;
```

```
SQL> ALTER DATABASE OPEN;
```

进行日志切换测试:

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

```
[oracle@rac11g1 ~]$ cd /oracle/archive/sky
```

```
[oracle@rac11g1 sky]$ ls -la *
```

```
-rw-r----- 1 oracle asmadmin 31943680 Oct 14 13:53 1_10_892990981.dbf
```

11.2.3.3 创建 STANDBY LOG (Primary Database)

REDO 信息:

```
SQL> SELECT GROUP#, BYTES FROM V$LOG;
```

GROUP#	BYTES
1	52428800
2	52428800
3	52428800

```
SQL> SELECT MEMBER FROM V$LOGFILE;
```

```
MEMBER
```

```
-----
```

```
/oracle/app/oracle/oradata/sky/redo03.log
```

```
/oracle/app/oracle/oradata/sky/redo02.log
```

```
/oracle/app/oracle/oradata/sky/redo01.log
```

创建 STANDBY LOG:

```
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/app/oracle/oradata/sky/skystandby1.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/app/oracle/oradata/sky/skystandby2.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/app/oracle/oradata/sky/skystandby3.log') SIZE 50M;
```

```
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/app/oracle/oradata/sky/skystandby4.log') SIZE 50M;
```

检查 STANDBY LOG:

```
SQL> SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

GROUP#	BYTES

4	52428800
5	52428800
6	52428800
7	52428800

```
SQL> SELECT GROUP#, TYPE, MEMBER FROM V$LOGFILE WHERE TYPE =
'STANDBY';
```

GROUP#	TYPE	MEMBER

4	STANDBY	/oracle/app/oracle/oradata/sky/skystandby1.log
5	STANDBY	/oracle/app/oracle/oradata/sky/skystandby2.log
6	STANDBY	/oracle/app/oracle/oradata/sky/skystandby3.log
7	STANDBY	/oracle/app/oracle/oradata/sky/skystandby4.log

11. 2. 3. 4 配置 Oracle Net Service Name (Primary/Standby/Cascade Database)

主数据库和备用数据库的 `tnsnames.ora` 文件都需要修改。

```
[oracle@rac11g1 ~]$ cd $ORACLE_HOME/network/admin
```

```
[oracle@rac11g1 admin]$ vi tnsnames.ora
```

```
CATALOG =
```

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g1)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = catalog)  
  )  
)
```

```
SKYPRIM =
```

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g1)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = skyprim)  
  )  
)
```

```
SKYSTAND =
```

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g2)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = skystand)  
  )  
)
```

```
SKYCASE =
```

```
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g2)(PORT = 1521))
```

```
(CONNECT_DATA =  
  (SERVER = DEDICATED)  
  (SERVICE_NAME = skycase)  
)  
)
```

➤ 节点 1 进行主机连接验证

```
[oracle@rac11g1 admin]$ tnsping sky
```

```
[oracle@rac11g1 admin]$ tnsping skystand
```

```
[oracle@rac11g1 admin]$ tnsping skycase
```

```
[oracle@rac11g1 admin]$ scp tnsnames.ora rac11g2:$ORACLE_HOME/network/admin
```

➤ 节点 2 进行主机连接验证

```
[oracle@rac11g2 admin]$ tnsping sky
```

```
[oracle@rac11g2 admin]$ tnsping skystand
```

```
[oracle@rac11g2 admin]$ tnsping skycase
```

由于 11g RAC 环境中，LISTENER 程序在 grid 用户下运行，默认情况下已经启动。所有 tnsnames.ora 文件配置正确的情况下，可以直接连接测试成功。

11.2.4 配置数据库静态监听

需要注意，11G 使用 ASM 的时候，监听文件在 grid 用户下。

使用 RMAN 的 duplicate 命令创建备库，备库必须首先处于 NOMOUNT 状态。在 NOMOUNT 状态下，数据库实例不会自动注册监听，必须配置静态监听。

➤ 主库

```
[oracle@rac11g1 admin]$ su - grid
```

```
[grid@rac11g1 ~]$ cd $ORACLE_HOME/network/admin
```

```
[grid@rac11g1 admin]$ vi listener.ora
```

```
--新增内容
```

```
SID_LIST_LISTENER =
```



```
(SID_LIST =  
  (SID_DESC =  
    (GLOBAL_DBNAME = sky)  
    (ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db_1)  
    (SID_NAME = sky)  
  )  
  (SID_DESC =  
    (GLOBAL_DBNAME = catalog)  
    (ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db_1)  
    (SID_NAME = catalog)  
  )  
)
```

➤ 备库

```
[oracle@rac11g2 admin]$ su - grid
```

```
[grid@rac11g2 ~]$ cd $ORACLE_HOME/network/admin
```

```
[grid@rac11g2 admin]$ vi listener.ora
```

--新增内容

```
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC =  
      (GLOBAL_DBNAME = skystand)  
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db_1)  
      (SID_NAME = skystand)  
    )  
    (SID_DESC =  
      (GLOBAL_DBNAME = skycase)  
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db_1)  
      (SID_NAME = skycase)  
    )  
  )
```

➤ 重新启动监听加载配置

```
[grid@rac11g1 admin]$ lsnrctl reload
```

```
[grid@rac11g1 admin]$ lsnrctl status
```

```
Services Summary...
```

```
Service "catalog" has 1 instance(s).
```

```
Instance "catalog", status UNKNOWN, has 1 handler(s) for this service...
```

```
Service "sky" has 1 instance(s).
```

```
Instance "sky", status UNKNOWN, has 1 handler(s) for this service...
```

```
The command completed successfully
```

```
[grid@rac11g1 admin]$ lsnrctl reload
```

```
[grid@rac11g1 admin]$ lsnrctl status
```

```
Services Summary...
```

```
Service "skycase" has 1 instance(s).
```

```
Instance "skycase", status UNKNOWN, has 1 handler(s) for this service...
```

```
Service "skystand" has 1 instance(s).
```

```
Instance "skystand", status UNKNOWN, has 1 handler(s) for this service...
```

```
The command completed successfully
```

11.2.5 数据库参数配置

11.2.5.1 创建 PFILE 文件

```
SQL> create pfile='/home/oracle/sky.pfile' from spfile;
```

```
SQL> shutdown immediate;
```

11.2.5.2 Primary Database

```
[oracle@rac11g1 ~]$ vi sky.pfile
```

```
audit_file_dest='/oracle/app/oracle/admin/sky/adump'
```

```
audit_trail='db'
```

```
compatible='11.2.0.4.0'
```

```
control_files='/oracle/app/oracle/oradata/sky/control01.ctl','/oracle/app/oracle/oradata/sky/control02.ctl'
```

```
db_block_size=8192
```

```
db_domain=""
diagnostic_dest='/oracle/app/oracle'
open_cursors=300
pga_aggregate_target=104857600
processes=150
sga_target=314572800
undo_tablespace='UNDOTBS1'

DB_NAME='sky'
DB_UNIQUE_NAME='SKYPRIM'
REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE'

LOG_ARCHIVE_CONFIG='DG_CONFIG=(SKYPRIM,SKYSTAND,SKYCASE)'
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/archive/skyprim/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SKYPRIM'
LOG_ARCHIVE_DEST_2='SERVICE=SKYSTAND SYNC AFFIRM
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=SKYSTAND'
LOG_ARCHIVE_DEST_3='SERVICE=SKYCASE
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE)
DB_UNIQUE_NAME=SKYCASE'

FAL_SERVER=SKYSTAND
DB_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/skystand/', '/oracle/app/oracle/o
radata/sky/'
LOG_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/skystand/', '/oracle/app/oracle/
oradata/sky/'

STANDBY_FILE_MANAGEMENT=AUTO
```

11. 2. 5. 3 Physical Standby Database

```
[oracle@rac11g1 ~]$ scp sky.pfile rac11g2:/home/oracle
[oracle@rac11g2 ~]$ cp sky.pfile skystand.pfile
[oracle@rac11g2 ~]$ vi skystand.pfile
audit_file_dest='/oracle/app/oracle/admin/skystand/adump'
audit_trail='db'
```

```
compatible='11.2.0.4.0'
control_files='/oracle/app/oracle/oradata/skystand/control01.ctl','/oracle/app/oracle/oradata/skystand/control02.ctl'
db_block_size=8192
db_domain=""
diagnostic_dest='/oracle/app/oracle'
open_cursors=300
pga_aggregate_target=104857600
processes=150
sga_target=314572800
undo_tablespace='UNDOTBS1'

DB_NAME='sky'
DB_UNIQUE_NAME='SKYSTAND'
REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE'

LOG_ARCHIVE_CONFIG='DG_CONFIG=(SKYPRIM,SKYSTAND,SKYCASE)'
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/archive/skystand/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SKYSTAND'
LOG_ARCHIVE_DEST_2='SERVICE=SKYPRIM SYNC AFFIRM
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=SKYPRIM'
LOG_ARCHIVE_DEST_3='SERVICE=SKYCASE ASYNC DELAY=10
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE)
DB_UNIQUE_NAME=SKYCASE'

FAL_SERVER=SKYPRIM
DB_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky','/oracle/app/oracle/oradata/skystand/'
LOG_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky','/oracle/app/oracle/oradata/skystand/'

STANDBY_FILE_MANAGEMENT=AUTO
```

11.2.5.4 Cascade Standby Database

```
[oracle@rac11g2 ~]$ vi skycase.pfile
```

```
audit_file_dest='/oracle/app/oracle/admin/skycase/adump'
audit_trail='db'
compatible='11.2.0.4.0'
control_files='/oracle/app/oracle/oradata/skycase/control01.ctl','/oracle/app/oracle/oradata/skycase/control02.ctl'
db_block_size=8192
db_domain=""
diagnostic_dest='/oracle/app/oracle'
open_cursors=300
pga_aggregate_target=104857600
processes=150
sga_target=314572800
undo_tablespace='UNDOTBS1'

DB_NAME='sky'
DB_UNIQUE_NAME='SKYCASE'
REMOTE_LOGIN_PASSWORDFILE='EXCLUSIVE'

LOG_ARCHIVE_CONFIG='DG_CONFIG=(SKYPRIM,SKYSTAND,SKYCASE)'
LOG_ARCHIVE_DEST_1='LOCATION=/oracle/archive/skycase/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SKYCASE'
LOG_ARCHIVE_DEST_2='SERVICE=SKYSTAND
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=SKYSTAND'

FAL_SERVER='SKYPRIM','SKYSTAND'
DB_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky','/oracle/app/oracle/oradata/skycase/'
LOG_FILE_NAME_CONVERT='/oracle/app/oracle/oradata/sky','/oracle/app/oracle/oradata/skycase/'

日志转换，如果不是主库的，rman 会失败，但是上面的日志传输信息确实第一个物理备库的。

STANDBY_FILE_MANAGEMENT=AUTO
```

级联备库的参数应该如何设置？

11.2.5.5 参数配置说明

11.2.6 复制 Primary Database 口令文件 (Primary Database)

```
[oracle@rac11g1 ~]$ cd $ORACLE_HOME/dbs  
[oracle@rac11g1 dbs]$ scp orapwsky rac11g2:$ORACLE_HOME/dbs/orapwskystand  
[oracle@rac11g1 dbs]$ scp orapwsky rac11g2:$ORACLE_HOME/dbs/orapwskycase
```

11.2.7 创建 Standby Database 所需目录

11.2.7.1 Physical Standby Database

```
[oracle@rac11g2 ~]$ mkdir -p /oracle/app/oracle/admin/skystand/adump  
[oracle@rac11g2 ~]$ mkdir -p /oracle/app/oracle/oradata/skystand/  
[oracle@rac11g2 ~]$ mkdir -p /oracle/archive/skystand/
```

11.2.7.2 Cascade Standby Database

```
[oracle@rac11g2 ~]$ mkdir -p /oracle/app/oracle/admin/skycase/adump  
[oracle@rac11g2 ~]$ mkdir -p /oracle/app/oracle/oradata/skycase/  
[oracle@rac11g2 ~]$ mkdir -p /oracle/archive/skycase/
```

11.2.8 启动 Standby Database

11.2.8.1 Physical Standby Database

```
[oracle@rac11g2 ~]$ export ORACLE_SID=skystand  
[oracle@rac11g2 ~]$ sqlplus / as sysdba  
SQL> startup nomount pfile='/home/oracle/skystand.pfile';  
SQL> create spfile from pfile='/home/oracle/skystand.pfile';  
SQL> shutdown immediate;  
SQL> startup nomount;
```

11.2.8.2 Cascade Standby Database

```
[oracle@rac11g2 ~]$ export ORACLE_SID=skycase
```

```
[oracle@rac11g2 ~]$ sqlplus / as sysdba  
SQL> startup nomount pfile='/home/oracle/skycase.pfile';  
SQL> create spfile from pfile='/home/oracle/skycase.pfile';  
SQL> shutdown immediate;  
SQL> startup nomount;
```

11. 2. 9 使用 RMAN 执行 DUPLICATE (Primary Database)

```
[oracle@rac11g1 ~]$ vi create_standby.rman
```

```
run  
{  
allocate channel prmy1 type disk;  
allocate channel prmy2 type disk;  
allocate channel prmy3 type disk;  
allocate channel prmy4 type disk;  
allocate auxiliary channel stand1 type disk;  
duplicate target database for standby from active database;  
}
```

11. 2. 9. 1 Physical Standby Database

```
[oracle@rac11g1 ~]$ rman  
RMAN> connect target sys/oracle@skyprim  
RMAN> connect auxiliary sys/oracle@skystand  
RMAN> @create_standby.rman
```

```
[oracle@rac11g2 ~]$ export ORACLE_SID=skystand
```

11. 2. 9. 2 Cascade Standby Database

```
[oracle@rac11g1 ~]$ rman  
RMAN> connect target sys/oracle@skyprim  
RMAN> connect auxiliary sys/oracle@skycase  
RMAN> @create_standby.rman
```

第一次报错：

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of Duplicate Db command at 10/14/2015 22:58:01
RMAN-05501: aborting duplication of target database
RMAN-05001: auxiliary file name /oracle/app/oracle/oradata/sky/users01.dbf conflicts with a file used by
the target database
RMAN-05001: auxiliary file name /oracle/app/oracle/oradata/sky/undotbs01.dbf conflicts with a file used
by the target database
RMAN-05001: auxiliary file name /oracle/app/oracle/oradata/sky/sysaux01.dbf conflicts with a file used
by the target database
RMAN-05001: auxiliary file name /oracle/app/oracle/oradata/sky/system01.dbf conflicts with a file used
by the target database
```

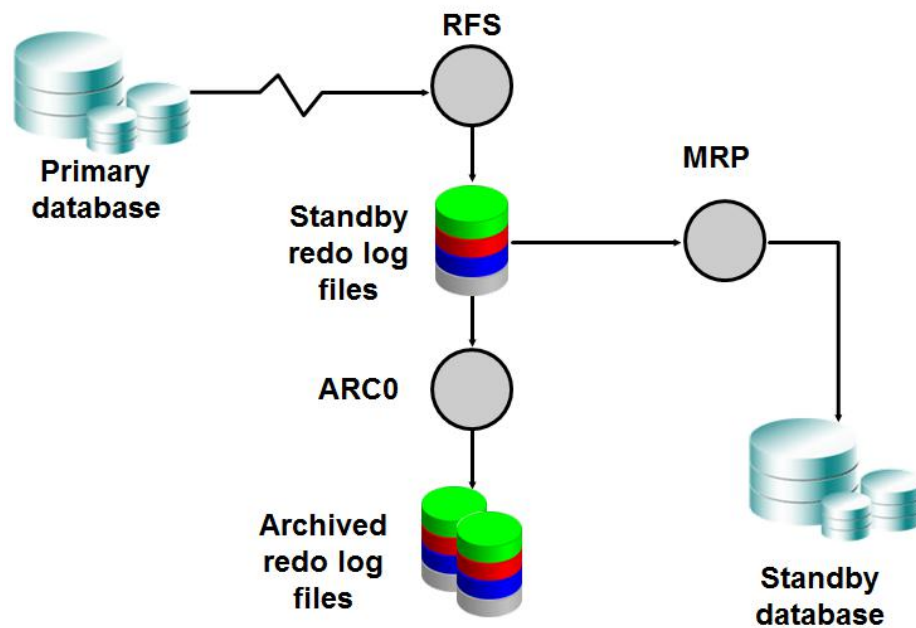
第二次报错:

```
using target database control file instead of recovery catalog
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-00601: fatal error in recovery manager
RMAN-03010: fatal error during library cache pre-loading
RMAN-10038: database session for channel clone_default terminated unexpectedly
```

第三次报错:

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of Duplicate Db command at 10/14/2015 23:17:20
RMAN-05501: aborting duplication of target database
RMAN-03015: error occurred in stored script Memory Script
RMAN-03009: failure of backup command on prmy4 channel at 10/14/2015 23:17:19
ORA-17628: Oracle error 19505 returned by remote Oracle server
```


11.2.10 启用 Real-Time Apply (Standby Database)

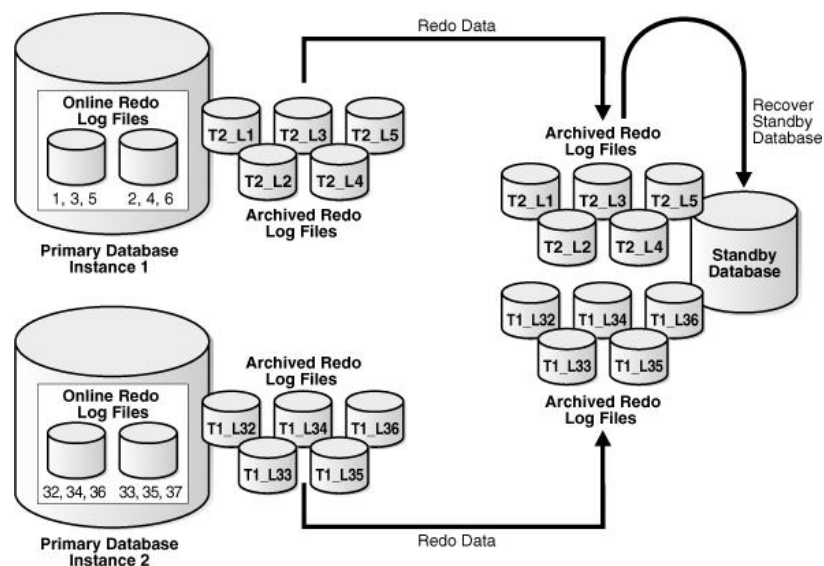


```
SQL> ALTER DATABASE OPEN;
```

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING  
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

11.3 RAC+DG(单实例)

11.3.1 环境信息规划



The standby database automatically determines the correct order in which to apply the archived redo log files.

Database	HOST	DB_UNIQUE_NAME	Net Service Name	DB_NAME
主库	rac11g1	acct	acct	acct
	rac11g2	acct	acct	
资料库	rac11g1	acctphy	acctphy	acct

11.3.2 RAC 主库准备

11.3.2.1 启动归档模式

- 单个节点执行

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=+DATA_DG/';
```

- 关闭所有节点

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP MOUNT;
```

- 单个节点执行

```
SQL> ALTER DATABASE ARCHIVELOG;
```

```
SQL> ALTER DATABASE OPEN;
```

- 启动其他节点

```
SQL> STARTUP;
```

- 进行日志切换测试

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

- 通过告警日志查看各节点日志切换情况

```
[oracle@rac11g1 ~]$ adrci
```

```
adrci> set home diag/rdbms/acct/acct1
```

```
adrci> show alert -tail -f
```

```

.....
2015-10-20 10:04:40.444000 +08:00
ALTER SYSTEM ARCHIVE LOG
2015-10-20 10:04:42.579000 +08:00
Thread 1 advanced to log sequence 10 (LGWR switch)
  Current log# 2 seq# 10 mem# 0: +DATA_DG/acct/onlineelog/group_2.304.893580637
.....

```

11.3.2.2 开启数据库 FORCE LOGGING

➤ 查看 FORCE LOGGING 模式

```
SQL> SELECT NAME, LOG_MODE, FORCE_LOGGING FROM GV$DATABASE;
```

NAME	LOG_MODE	FOR
ACCT	ARCHIVELOG	NO
ACCT	ARCHIVELOG	NO

➤ 开启 FORCE LOGGING 模式

```
SQL> ALTER DATABASE FORCE LOGGING;
```

➤ 查看 FORCE LOGGING 模式

```
SQL> SELECT NAME, LOG_MODE, FORCE_LOGGING FROM GV$DATABASE;
```

NAME	LOG_MODE	FOR
ACCT	ARCHIVELOG	YES
ACCT	ARCHIVELOG	YES

➤ 关闭 FORCE LOGGING 模式

```
SQL> ALTER DATABASE NO FORCE LOGGING;
```

11.3.2.3 创建 STANDBY LOG

➤ REDO 信息

SQL> SELECT GROUP#, THREAD#, BYTES FROM V\$LOG;

GROUP#	THREAD#	BYTES
1	1	52428800
2	1	52428800
3	2	52428800
4	2	52428800

SQL> SELECT GROUP#, MEMBER FROM V\$LOGFILE ORDER BY 1;

GROUP#	MEMBER
1	+DATA_DG/acct/onlineolog/group_1.302.893580635
2	+DATA_DG/acct/onlineolog/group_2.304.893580637
3	+DATA_DG/acct/onlineolog/group_3.327.893581327
4	+DATA_DG/acct/onlineolog/group_4.322.893581331

➤ 创建 STANDBY LOG

由于主库为 RAC 环境，因此每个实例都需要进行日志的添加操作。STANDBY 日志数为在线日志数+1。

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 5 SIZE 50M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 6 SIZE 50M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 7 SIZE 50M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 8 SIZE 50M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 9 SIZE 50M;

ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 10 SIZE 50M;

➤ 检查 STANDBY LOG

SQL> SELECT GROUP#, BYTES FROM V\$STANDBY_LOG;

GROUP#	BYTES
5	52428800
6	52428800
7	52428800

```
8 52428800
9 52428800
10 52428800
```

```
SQL> SELECT GROUP#, TYPE, MEMBER FROM V$LOGFILE WHERE TYPE =
'STANDBY' ORDER BY 1;
```

GROUP#	TYPE	MEMBER
5	STANDBY	+DATA_DG/acct/onlineelog/group_5.287.893586985
6	STANDBY	+DATA_DG/acct/onlineelog/group_6.294.893587015
7	STANDBY	+DATA_DG/acct/onlineelog/group_7.277.893587019
8	STANDBY	+DATA_DG/acct/onlineelog/group_8.286.893587021
9	STANDBY	+DATA_DG/acct/onlineelog/group_9.312.893587025
10	STANDBY	+DATA_DG/acct/onlineelog/group_10.276.893587027

11.3.2.4 主库备份

➤ 备份数据文件

```
RMAN>
```

```
run
```

```
{
```

```
allocate channel c1 type disk;
```

```
allocate channel c2 type disk;
```

```
allocate channel c3 type disk;
```

```
allocate channel c4 type disk;
```

```
backup database format '/oracle/rman/ACCT_FULL_%U.bak';
```

```
release channel c1;
```

```
release channel c2;
```

```
release channel c3;
```

```
release channel c4;
```

```
}
```

➤ 备份归档日志

```
RMAN> backup archivelog all format '/oracle/rman/ACCT_ARCH_%U.bak';
```

➤ 备份备库控制文件

RMAN>

```
backup device type disk format '/oracle/rman/ACCT_STANDBY_%U.ctl' CURRENT  
CONTROLFILE FOR STANDBY;
```

进行备库控制文件备份时，ORACLE 会提示进行创建 STANDBY 日志的操作

2015-10-20 10:22:52.374000 +08:00

Clearing standby activation ID 3830669012 (0xe4535ed4)

The primary database controlfile was created using the
'MAXLOGFILES 192' clause.

There is space for up to 188 standby redo logfiles

Use the following SQL commands on the standby database to create
standby redo logfiles that match the primary database:

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr1.f' SIZE 52428800;
```

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr2.f' SIZE 52428800;
```

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr3.f' SIZE 52428800;
```

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr4.f' SIZE 52428800;
```

```
ALTER DATABASE ADD STANDBY LOGFILE 'sr5.f' SIZE 52428800;
```

WARNING: OMF is enabled on this database. Creating a physical
standby controlfile, when OMF is enabled on the primary
database, requires manual RMAN intervention to resolve OMF
datafile pathnames.

NOTE: Please refer to the RMAN documentation for procedures
describing how to manually resolve OMF datafile pathnames.

➤ 备份参数文件

```
SQL> CREATE PFILE='/home/oracle/acct.pfile' FROM SPFILE;
```

➤ 调整主库口令文件

保证所有节点口令文件中 sys 口令一致。

11.3.2.5 修改主库参数文件

```
ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(ACCT,ACCTPHY)';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=+DATA_DG VALID_FOR=(ALL_LOGFILES,ALL_ROLES)  
DB_UNIQUE_NAME=ACCT';
```

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=ACCTPHY  
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=ACCTPHY';
```

主备均做参数

```
ALTER SYSTEM SET FAL_SERVER=ACCTPHY;  
ALTER SYSTEM SET DB_FILE_NAME_CONVERT='+DATA_DG/acctphy/','+DATA_DG/acct/' scope=spfile;  
ALTER SYSTEM SET LOG_FILE_NAME_CONVERT='+DATA_DG/acctphy/','+DATA_DG/acct/' scope=spfile;  
  
ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT='AUTO';
```

由于主库使用共享存储，因此使用 **CREATE SPFILE** 方式重建的参数文件，会直接创建到本地。因此建议使用 **ALTER SYSTEM** 的方式进行参数的修改。或 **CREATE SPFILE='+DATA_DG/acct/spfileacct.ora' FROM PFILE='/home/oracle/acct.pfile'** 方式进行修改。

11.3.2.6 重启数据库验证参数

检查告警日志中是否有报错信息。

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP;
```

11.3.2.7 修改 TNS 连接

```
[oracle@rac11g1]$ cd $ORACLE_HOME/network/admin
```

```
[oracle@rac11g1 admin]$ vi tnsnames.ora
```

--新增内容

```
ACCT =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g-scan)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = acct)  
    )  
  )
```

```
ACCTPHY =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g2)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = acctphy)  
    )  
  )
```

11.3.3 物理备库准备

11.3.3.1 拷贝备份到备库

```
[oracle@rac11g1 oracle]$ scp -r ./rman/ rac11g2:/oracle
```

11.3.3.2 创建备库口令文件

使用主库的口令文件，设置备库的口令文件。新建口令文件的方式容易出现主备库口令不一致，日志传输失败的问题。

```
[oracle@rac11g2 ~]$ cd $ORACLE_HOME/dbs  
[oracle@rac11g2 dbs]$ cp orapwacct2 orapwacctphy
```

11.3.3.3 修改备库初始化参数文件

```
[oracle@rac11g1 ~]$ scp /home/oracle/acct.pfile rac11g2:/home/oracle  
[oracle@rac11g2 ~]$ mv acct.pfile acctphy.pfile  
[oracle@rac11g2 ~]$ cat acctphy.pfile
```

取消 RAC 相关的参数设置。

```
audit_file_dest='/oracle/app/oracle/admin/acctphy/adump'  
audit_trail='db'  
compatible='11.2.0.4.0'  
db_block_size=8192  
db_domain=""  
diagnostic_dest='/oracle/app/oracle'  
open_cursors=300  
pga_aggregate_target=209715200  
processes=150  
remote_login_passwordfile='exclusive'  
sga_target=419430400  
undo_tablespace='UNDOTBS1'  
  
db_name='acct'  
DB_UNIQUE_NAME='acctphy'
```

--备库采用 ASM 存储方式，因此指定 db_create_file_dest 目录即可，控制文件的路径需要指定
control_files='+DATA_DG/acctphy/controlfile/current.ctl'


```
db_create_file_dest='+DATA_DG'

LOG_ARCHIVE_CONFIG='DG_CONFIG=(ACCT,ACCTPHY)'
LOG_ARCHIVE_DEST_1='LOCATION='+DATA_DG VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=ACCTPHY'
LOG_ARCHIVE_DEST_2='SERVICE=ACCT VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=ACCT'

FAL_SERVER=ACCT
DB_FILE_NAME_CONVERT='+DATA_DG/acct','+DATA_DG/acctphy/'
LOG_FILE_NAME_CONVERT='+DATA_DG/acct','+DATA_DG/acctphy/'

STANDBY_FILE_MANAGEMENT=AUTO
```

11.3.3.4 创建相关目录

诊断日志的目录实例会自动创建，因此只需建 **adump** 目录即可。

```
[oracle@rac11g2 ~]$ mkdir -p /oracle/app/oracle/admin/acctphy/adump
```

11.3.3.5 配置 TNS 连接

```
[oracle@rac11g2 ~]$ cd $ORACLE_HOME/network/admin
```

```
[oracle@rac11g2 admin]$ vi tnsnames.ora
```

--新增内容

```
ACCT =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g-scan)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = acct)
    )
  )

ACCTPHY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac11g2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = acctphy)
    )
  )
```

```
)  
)
```

11.3.4 创建物理备库

11.3.4.1 启动备库到 NOMOUNT

指定备库的 ORACLE_SID，进行实例启动操作。

```
[oracle@rac11g2 ~]$ export ORACLE_SID=acctphy  
SQL> startup nomount pfile='/home/oracle/acctphy.pfile';  
SQL> ! ps -ef | grep ora_smon
```

oracle	15748	1	0	11:36 ?	00:00:00	ora_smon_acct2
oracle	16289	1	0	11:41 ?	00:00:00	ora_smon_acctphy

由于备库为单实例，直接创建本地 spfile 文件即可。

```
SQL> create spfile from pfile='/home/oracle/acctphy.pfile';
```

11.3.4.2 恢复控制文件

指定备库的 ORACLE_SID，进行 RMAN 操作。

```
[oracle@rac11g2 rman]$ export ORACLE_SID=acctphy  
RMAN>  
restore standby controlfile from '/oracle/rman/ACCT_STANDBY_0eqk6424_1_1.ctl';
```

11.3.4.3 MOUNT 数据库

将备库的状态修改为 MOUNT。

```
[oracle@rac11g2 ~]$ export ORACLE_SID=acctphy  
SQL> ALTER DATABASE MOUNT;
```

11.3.4.4 还原数据库

主库的备份信息直接记录在控制文件中，因此恢复控制文件后，就可以看到数据库的备份信息。

```
[oracle@rac11g2 rman]$ export ORACLE_SID=acctphy  
[oracle@rac11g2 rman]$ rman target /  
RMAN> LIST BACKUP;
```

RMAN>

```
run
{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
allocate channel c4 type disk;
restore database;
release channel c1;
release channel c2;
release channel c3;
release channel c4;
}
```

11.3.4.5 日志切换测试

➤ 日志同步错误信息

Error 1017 received logging on to the standby

Check that the primary and standby are using a password file
and remote_login_passwordfile is set to SHARED or EXCLUSIVE,
and that the SYS password is same in the password files.

returning error ORA-16191

FAL[client, MRP0]: Error 16191 connecting to ACCT for fetching gap sequence

2015-10-20 11:50:47.218000 +08:00

Error 1033 received logging on to the standby

FAL[server, ARC3]: Error 1033 creating remote archivelog file 'ACCTPHY'

FAL[server, ARC3]: FAL archive failed, see trace file.

ARCH: FAL archive failed. Archiver continuing

ORACLE Instance acct1 - Archival Error. Archiver continuing.

该错误与主库和备库的 **sys** 口令不一致导致。虽然手工设置为一致，但是仍然会有这种错误，直接的解决办法是用主库的口令文件重新同步备库的口令文件。

```
[oracle@rac11g2 ~]$ cd $ORACLE_HOME/dbs
```

```
[oracle@rac11g2 dbs]$ cp orapwacct2 orapwacctphy
```

11.3.5 打开备用数据库

11.3.5.1 启用日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

11.3.5.2 取消日志应用

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

11.3.5.3 RTA 方式打开备库

由于备份的数据库处于不一致的状态，因此在打开数据库之前，首先要做日志的应用。日志应用完成后，取消日志应用操作，然后将数据库至于打开状态。

```
SQL> ALTER DATABASE OPEN;
```

11.3.6 RAC 主库 SWITCHOVER 单机备库

11.3.6.1 RAC 主库执行切换备库操作

➤ 检查主库的状态

```
SQL> SELECT INST_ID, DBID, NAME, PROTECTION_MODE, DATABASE_ROLE,
FORCE_LOGGING, OPEN_MODE, SWITCHOVER_STATUS FROM GV$DATABASE;
```

INST_ID	DBID	NAME	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
1	3830711767	ACCT	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	SESSIONS ACTIVE
2	3830711767	ACCT	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	SESSIONS ACTIVE

主库状态为 TO STANDBY 或 SESSIONS ACTIVE 表示主库可以进行切换备库操作。

➤ 只保留一个 RAC 实例

```
SQL> SHUTDOWN IMMEDIATE;
```

■ 检查主库状态

INST_ID	DBID	NAME	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
1	3830711767	ACCT	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	TO STANDBY

保留实例状态为 TO STANDBY。

■ 检查备库状态

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1189720	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES READ ONLY WITH APPLY	NOT ALLOWED

➤ 主库保留节点执行切换操作

SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY WITH SESSION SHUTDOWN;

➤ MOUNT 原 RAC 主库

[oracle@rac11g1 ~]\$ srvctl start database -d acct -o mount

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1209778	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES MOUNTED	RECOVERY NEEDED
3830711767	ACCT	1209778	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES MOUNTED	RECOVERY NEEDED

11.3.6.2 单节点备库执行切换主库操作

➤ 检查备库状态

SQL> SELECT DBID, NAME, CURRENT_SCN, PROTECTION_MODE, DATABASE_ROLE, FORCE_LOGGING, OPEN_MODE, SWITCHOVER_STATUS FROM GV\$DATABASE;

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1209779	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES READ ONLY WITH APPLY	TO PRIMARY

➤ 切换物理备库到主库角色

SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WITH SESSION SHUTDOWN;

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	0	MAXIMUM PERFORMANCE	PRIMARY	YES MOUNTED	NOT ALLOWED

➤ 打开单节点新主库

SQL> ALTER DATABASE OPEN;

SQL> SELECT DBID, NAME, CURRENT_SCN, PROTECTION_MODE, DATABASE_ROLE, FORCE_LOGGING, OPEN_MODE, SWITCHOVER_STATUS FROM GV\$DATABASE;

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1161046	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	FAILED DESTINATION

11.3.6.3 RAC 备库启动日志恢复

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT FROM SESSION;

新 RAC 备库状态

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1209828	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES MOUNTED	NOT ALLOWED
3830711767	ACCT	1209828	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES MOUNTED	NOT ALLOWED

11.3.6.4 新主库进行日志切换测试

➤ 打开新 RAC 备库

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

SQL> ALTER DATABASE OPEN;

启用实时日志恢复

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT FROM SESSION;

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1210517	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES READ ONLY WITH APPLY	NOT ALLOWED
3830711767	ACCT	1210517	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES READ ONLY WITH APPLY	NOT ALLOWED

➤ 新主库创建测试对象

SQL> CREATE TABLE SKY (ID NUMBER);

SQL> ALTER SYSTEM SWITCH LOGFILE;

➤ 新备库节点进行查询

■ 节点 1

SQL> SELECT * FROM SKY;

■ 节点 2

```
SQL> SELECT * FROM SKY;
```

11.3.7 新单节点主库 SWITCHOVER 旧 RAC 备库

11.3.7.1 停旧 RAC 备库节点

保留一个节点用作切换

```
SQL> SELECT DBID, NAME, CURRENT_SCN, PROTECTION_MODE, DATABASE_ROLE,
FORCE_LOGGING, OPEN_MODE, SWITCHOVER_STATUS FROM GV$DATABASE;
```

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR	OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1210913	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES	READ ONLY WITH APPLY	NOT ALLOWED
3830711767	ACCT	1210913	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES	READ ONLY WITH APPLY	NOT ALLOWED

```
SQL> SHUTDOWN IMMEDIATE;
```

11.3.7.2 新单机主库切换到备库

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY WITH SESSION
SHUTDOWN;
```

11.3.7.3 新 RAC 单节点备库切换为主库

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR	OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1231103	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES	READ ONLY WITH APPLY TO PRIMARY	

```
SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WITH SESSION
SHUTDOWN;
```

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR	OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	0	MAXIMUM PERFORMANCE	PRIMARY	YES	MOUNTED	NOT ALLOWED

11.3.7.4 打开新 RAC 主库所有实例

➤ 保留节点

```
SQL> ALTER DATABASE OPEN;
```

➤ 其他节点

SQL> **STARTUP;**

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1233963	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	SESSIONS ACTIVE
3830711767	ACCT	1233963	MAXIMUM PERFORMANCE	PRIMARY	YES READ WRITE	SESSIONS ACTIVE

11.3.7.5 新单节点备库启动并应用日志

SQL> **STARTUP;**

SQL> **ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE **PARALLEL 4** DISCONNECT FROM SESSION;**

通过并行，加速日志的恢复。

DBID	NAME	CURRENT_SCN	PROTECTION_MODE	DATABASE_ROLE	FOR OPEN_MODE	SWITCHOVER_STATUS
3830711767	ACCT	1233930	MAXIMUM PERFORMANCE	PHYSICAL STANDBY	YES READ ONLY WITH APPLY NOT ALLOWED	

11.3.7.6 新主库进行日志切换测试

➤ 节点 1 测试

SQL> **INSERT INTO SKY VALUES (2);**

SQL> **COMMIT;**

➤ 节点 2 测试

SQL> **INSERT INTO SKY VALUES (3);**

SQL> **COMMIT;**

➤ 备库测试

SQL> **SELECT * FROM SKY;**

➤ 通过归档方式测试日志传输

SQL> **ALTER SYSTEM ARCHIVE LOG CURRENT;**

11.3.8 FAILOVER 测试

11.3.9 常见问题处理

11.4 RAC+DG(RAC)

12 故障处理

12.1 Manual Gap Resolution

```
SQL> SELECT * FROM V$ARCHIVE_GAP;
```

```
SQL> SELECT NAME FROM V$ARCHIVED_LOG  
WHERE THREAD#=1 AND DEST_ID=1 AND SEQUENCE# BETWEEN 7 AND 10;
```

Copy these log files to the physical standby database and register them using the ALTER DATABASE REGISTER LOGFILE.

```
SQL> ALTER DATABASE REGISTER LOGFILE  
'/physical_standby1/thread1_dest/arcr_1_7.arc';
```

注意事项

The V\$ARCHIVE_GAP view on a physical standby database only returns the gap that is currently blocking Redo Apply from continuing. After resolving the gap, query the V\$ARCHIVE_GAP view again on the physical standby database to determine if there is another gap sequence. Repeat this process until there are no more gaps.

12.2 ORA-10458: standby database requires recovery

备库异常崩溃后，重启备库到 OPEN 状态，会提示需要进行恢复。

```
ORA-10458: standby database requires recovery
```

```
ORA-01196: file 1 is inconsistent due to a failed media recovery session
```

```
ORA-01110: data file 1: '/oradata/testdb/system01.dbf'
```

告警日志中，

```
2018-08-15 02:26:31.672000 +08:00
```

```
Standby crash recovery failed to bring standby database to a consistent  
point because needed redo hasn't arrived yet.
```

```
MRP: Wait timeout: thread 1 sequence# 896
```

```
Standby Crash Recovery aborted due to error 16016.
```

```
Errors in file /oracle/app/oracle/diag/rdbms/testdb/testdb/trace/testdb_oracle_3250.trc:
ORA-16016: archived log for thread 1 sequence# 896 unavailable
Recovery interrupted!
Some recovered datafiles maybe left media fuzzy
Media recovery may continue but open resetlogs may fail
Completed Standby Crash Recovery.
Errors in file /oracle/app/oracle/diag/rdbms/testdb/testdb/trace/testdb_oracle_3250.trc:
ORA-10458: standby database requires recovery
ORA-01196: file 1 is inconsistent due to a failed media recovery session
ORA-01110: data file 1: '/oradata/testdb/system01.dbf'
ORA-10458 signalled during: ALTER DATABASE OPEN...
```

该问题是由于备库异常崩溃后，实例需要进行恢复，对数据库进行一次正常启停操作，故障解除。

```
SQL> shutdown immediate;
```

```
SQL> startup;
```

13 参考文献

11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus (Doc ID 1304939.1)

Redo Transport Compression in a Data Guard Environment (文档 ID 729551.1)

_REDO_TRANSPORT_COMPRESS_ALL=TRUE

Known issues with RMAN DUPLICATE FROM ACTIVE DATABASE (文档 ID 1366290.1)

ARCH ping 进程会查询备库控制文件(通过 RFS 进程)

坏块保护:

db_ultra_safe

db_lost_write_protect

db_block_checksum

```
SELECT * FROM V$SYSTEM_EVENT WHERE EVENT LIKE 'LNS%';
```

STANDBY_FILE_MANAGEMENT 参数修改为 MANUAL，可以进行 ORL 操作。