

Oracle® Tutor™



Section 8 Data Guard 注意事项和答案

1. 注意点

1. alter database force logging;
2. standby logfile lgwr async
3. dataguard 切换要保证网络正常切换到 standby ,运行一个脚本，然后在切换回，运行一个脚本。
4. read only 打开 standby
5. 确保库 open,能本地连接/网络连接

2. 考点

- 1、主库初始化参数文件的配置
- 2、备库初始化参数文件的配置
- 3、主备库角色的切换
- 4、主备库监听的配置
- 6、主备库 TNS 的配置

难点：初始化参数文件的配置

注意点：先备份，以防万一；建议使用 GC 配置。万一出问题，日志在 gc 的

\$ORACLE_BASE/OracleHomes/oms10g/sysman/log/。

在 gc 向导的第四步要给出文件对应关系这一步骤非常的关键，无数的仁人志士跌倒在这里。此处一定不要急于点击 Next 进入到下一环节，我们需要点击 Customize 对文件目录进行定制。否则 standby_file_management 参数将为“MANUAL”、db_file_name_convert 及 log_file_name_convert 参数为空，当主库添加数据文件后备库将无法恢复。

注意：单单的给出路径是不够的，一定要点击路径后面的“Go”。只有这样，给出的路径名才会真正的得到应用，否则无济于事。

1) 一一修改，确保修改全面后点击“OK”。

2) 点击 OK 后，会收到如下的提醒信息。该警告信息是由于我们指定的目录事先没有创建导致的，点击 Yes，系统将自动创建。

3) 又回到 File Locations 界面，似乎什么都没有发生，但什么事情都有所改变。点击 Next 继续。

2) 仔细检查主备数据库数据文件、临时文件、日志文件和控制文件的对应关系是否调整正确。一切确认无误后点击 Finish，此后将进入到无人值守的自动创建阶段。

2) 自动创建 Job 后台自动运行，此时可以点击“Creation in progress”跟踪整个 Job 的运行过程

如果一切顺利，使用 Grid Control 创建物理 Data Guard 大约仅需 10 分钟左右的时间，可谓速度惊人。在得到便捷的同时，该方法也隐藏了很多实现细节，不便于深入了解 Data Guard 的运行原理。要学习技术，建议以脚本创建为主 GC 创建为辅的原则来探索 Data Guard 的奥秘。

1) 会用 Maximum Protection、Maximum Availability、Maximum Performance 中的一种模式来配置 dataguard。一般 Maximum Availability 模式考的多一些。

2) dataguard 配置完成后，做如下要求：

第一，要求在主库上运行脚本 1 创建一个表并插入当前时间。

第二，要求做一次切换，使主备互换。在切换后，再运行脚本 2 在原备库上同一张表插入当前时间。

第三，要求再做一次切换，使互换后的主备重新切回来。再运行脚本 3 在主库上同一张表插入当前时间。

3) 只考物理 standby。

4) 要求 read only 打开 standby

5) 考 switchover。

3.主要流程

搭建一个 Data Guard 在最大保护模式下进行两次切换，在切换前跑相应脚本：

1. 设置 LISTENER，把备库(PRODSTD)也设置进默认 LISTENER 里面
SID_LIST_LISTENER =

```

(SID_LIST =
  (SID_DESC =
    (GLOBAL_DBNAME = emrep)
    (ORACLE_HOME = /home/oracle/product/10.2.0/db_1)
    (SID_NAME = emrep)
  )
  (SID_DESC =
    (GLOBAL_DBNAME = prod)
    (ORACLE_HOME = /home/oracle/product/10.2.0/db_1)
    (SID_NAME = prod)
  )
  (SID_DESC =
    (GLOBAL_DBNAME = prodstd)
    (ORACLE_HOME = /home/oracle/product/10.2.0/db_1)
    (SID_NAME = prodstd)
  )
)
)
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = celty.localdomain)(PORT = 1521))
  )
)

```

2. 数据库必须是归档状态，FORCE LOGGING 状态

```

SQL> select FORCE_LOGGING from v$database;
        FORCE
        ---
        NO

```

```

SQL> archive log list;
Database log mode          Archive Mode
Automatic archival         Enabled
Archive destination        /home/oracle/oradata/prod/arc/
Oldest online log sequence 17
Next log sequence to archive 21
Current log sequence        21

```

3. 在 initPROD.ora 的尾部加入修改后的参数（参考文档 Data Guard Concepts and Administration 中的 3 Creating a Physical Standby Database）大致修改如下：

主库（PROD）：**删除一切关于 MTS 的参数（shared server、dispatcher），否则有可能数据库无法切换。**

```

DB_UNIQUE_NAME=PROD
LOG_ARCHIVE_CONFIG='DG_CONFIG=( PROD, PRODSTD)'
LOG_ARCHIVE_DEST_1=
  'LOCATION=/home/oracle/oradata/PROD/arc/

```

```
VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=PROD'
LOG_ARCHIVE_DEST_2=
'SERVICE= PRODSTD LGWR SYNC AFFIRM
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME= PRODSTD '
LOG_ARCHIVE_DEST_STATE_1=ENABLE
LOG_ARCHIVE_DEST_STATE_2=ENABLE
```

```
FAL_SERVER= PRODSTD
FAL_CLIENT=PROD
DB_FILE_NAME_CONVERT=' PRODSTD ','PROD'
LOG_FILE_NAME_CONVERT=' PRODSTD ',' PROD'
STANDBY_FILE_MANAGEMENT=AUTO
3.2 复制一份 initPROD.ora 改名为 initPRODSTD.ora
cd $ORACLE_HOME/dbs
cp initPROD.ora initPRODSTD.ora
3.3 修改 initPRODSTD.ora 的参数如下
备库 (PRODSTD)
```

```
DB_UNIQUE_NAME=PRODSTD
LOG_ARCHIVE_CONFIG='DG_CONFIG=( PRODSTD, PROD)'
LOG_ARCHIVE_DEST_1=
'LOCATION=/home/oracle/oradata/PRODSTD/arc/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=PRODSTD'
LOG_ARCHIVE_DEST_2=
'SERVICE= PROD LGWR SYNC AFFIRM
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME= PROD '
LOG_ARCHIVE_DEST_STATE_1=ENABLE
LOG_ARCHIVE_DEST_STATE_2=ENABLE
```

```
FAL_SERVER= PROD
FAL_CLIENT=PRODSTD
DB_FILE_NAME_CONVERT=' PROD','PRODSTD'
LOG_FILE_NAME_CONVERT=' PROD',' PRODSTD'
STANDBY_FILE_MANAGEMENT=AUTO
```

4 配置密码文件

```
cd $ORACLE_HOME/dbs
orapwd file=orapwprodstd password=oracle
```

5 PROD 打开至 MOUNT 状态，复制文件

```
[oracle@celty oradata]$ sqlplus / as sysdba
SQL*Plus: Release 10.2.0.1.0 - Production on Thu May 10 10:58:23 2012
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.
```

Total System Global Area 314572800 bytes

Fixed Size 1219184 bytes

Variable Size 96470416 bytes

Database Buffers 213909504 bytes

Redo Buffers 2973696 bytes

Database mounted.

SQL> !

```
[oracle@celty oradata]$ cp -R prod prodstd (复制全部文件)
```

1. 创建备库的控制文件

```
alter database create standby controlfile as
'/home/oracle/oradata/prodstd/control01.ctl'
要跟原来的控制文件相同数量
cp /home/oracle/oradata/prodstd/control01.ctl
/home/oracle/oradata/prodstd/control02.ctl
cp /home/oracle/oradata/prodstd/control01.ctl
/home/oracle/oradata/prodstd/control03.ctl
```

7. 将 PRODSTD 打开也进入 mount 状态

加入 STANDBY LOGFILE

在 PROD 中

Alter database add standby logfile

Group 4 '/home/oracle/oradata/prod/stdredo01.log' size 100M,

```
alter database add standby logfile group 5 '/u01/app/oradata/tdb/sredo5.dbf' size
100m;
```

```
alter database add standby logfile group 6 '/u01/app/oradata/tdb/sredo6.dbf' size
100m;
```

```
alter database add standby logfile group 7 '/u01/app/oradata/tdb/sredo7.dbf' size
100m;
```

在 PRODSTD 中

Alter database add standby logfile

Group 4 '/home/oracle/oradata/prodstd/stdredo01.log' size 100M,

...

(注：standby logfile 的大小必须与原本数据库的 logfile 大小相同，而且数量比原来多一个)

查询

Select name from v\$datafile;

Select member from v\$logfile;

看看是否已经转换成功

```
/home/oracle/oradata/prod/system01.dbf
/home/oracle/oradata/prod/undotbs01.dbf
/home/oracle/oradata/prod/sysaux01.dbf
/home/oracle/oradata/prod/example.dbf
/home/oracle/oradata/prod/indx.dbf
/home/oracle/oradata/prod/tools.dbf
/home/oracle/oradata/prod/users.dbf
/home/oracle/oradata/prod/oltp.dbf
/home/oracle/oradata/prod/REGISTRATION01.dbf
/home/oracle/oradata/prod/part1/lob_data01.dbf
/home/oracle/oradata/prod/part2/lob_data02.dbf
```

NAME

```
/home/oracle/oradata/prod/data01.dbf
/home/oracle/oradata/prod/data02.dbf
/home/oracle/oradata/prod/data03.dbf
/home/oracle/oradata/prod/data04.dbf
/home/oracle/oradata/prod/data05.dbf
```

16 rows selected.

```
/home/oracle/oradata/prodstd/system01.dbf
/home/oracle/oradata/prodstd/undotbs01.dbf
/home/oracle/oradata/prodstd/sysaux01.dbf
/home/oracle/oradata/prodstd/example.dbf
/home/oracle/oradata/prodstd/indx.dbf
/home/oracle/oradata/prodstd/tools.dbf
/home/oracle/oradata/prodstd/users.dbf
/home/oracle/oradata/prodstd/oltp.dbf
/home/oracle/oradata/prodstd/REGISTRATION01.dbf
/home/oracle/oradata/prodstd/part1/lob_data01.dbf
/home/oracle/oradata/prodstd/part2/lob_data02.dbf
```

NAME

```
/home/oracle/oradata/prodstd/data01.dbf
/home/oracle/oradata/prodstd/data02.dbf
/home/oracle/oradata/prodstd/data03.dbf
/home/oracle/oradata/prodstd/data04.dbf
/home/oracle/oradata/prodstd/data05.dbf
```

16 rows selected.

```
/home/oracle/oradata/prod/redo1/redo01a.log
/home/oracle/oradata/prod/redo2/redo01b.log
/home/oracle/oradata/prod/redo1/redo02a.log
/home/oracle/oradata/prod/redo2/redo02b.log
/home/oracle/oradata/prod/redo1/redo03a.log
/home/oracle/oradata/prod/redo2/redo03b.log
/home/oracle/oradata/prod/redo1/redo04a.log
/home/oracle/oradata/prod/redo2/redo04b.log
/home/oracle/oradata/prod/redo1/redo05a.log
/home/oracle/oradata/prod/redo2/redo05b.log
/home/oracle/oradata/prod/stdredo01.dbf
```

MEMBER

```
/home/oracle/oradata/prod/stdredo02.dbf
/home/oracle/oradata/prod/stdredo03.dbf
/home/oracle/oradata/prod/stdredo04.dbf
/home/oracle/oradata/prod/stdredo05.dbf
/home/oracle/oradata/prod/stdredo06.dbf
```

16 rows selected.

```
/home/oracle/oradata/prodstd/redo1/redo01a.log
/home/oracle/oradata/prodstd/redo2/redo01b.log
/home/oracle/oradata/prodstd/redo1/redo02a.log
/home/oracle/oradata/prodstd/redo2/redo02b.log
/home/oracle/oradata/prodstd/redo1/redo03a.log
/home/oracle/oradata/prodstd/redo2/redo03b.log
/home/oracle/oradata/prodstd/stdredo01.dbf
/home/oracle/oradata/prodstd/redo1/redo04a.log
/home/oracle/oradata/prodstd/redo2/redo04b.log
/home/oracle/oradata/prodstd/redo1/redo05a.log
/home/oracle/oradata/prodstd/redo2/redo05b.log
```

MEMBER

```
/home/oracle/oradata/prodstd/stdredo02.dbf
/home/oracle/oradata/prodstd/stdredo03.dbf
/home/oracle/oradata/prodstd/stdredo04.dbf
/home/oracle/oradata/prodstd/stdredo05.dbf
/home/oracle/oradata/prodstd/stdredo06.dbf
```

16 rows selected.

8. 此时将 PROD 切换至最大保护模式（在 MOUNT 状态才可以使用该命令）

SQL>ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PROTECTION

打开 PROD 数据库进入 OPEN 状态

跑脚本 q1.sql

SQL>@q1.sql

9. 在 PROD 进行切换

SQL>Alter database commit to switchover to physical standby

成功后立刻 shutdown immediate 然后再 startup mount

此时数据库 PROD 的 database_role 已经变成 physical standby

在 PRODSTD 经行一次恢复

SQL>Alter database recover managed standby database disconnect from session

等后台恢复结束后

Alter database commit to switchover to primary

此时 PRODSTD 的 database_role 已经变成 PRIMARY

运行脚本 q2.sql

SQL>@q2.sql

10. 在 PRODSTD 经行切换

SQL>Alter database commit to switchover to physical standby

成功后立刻 shutdown immediate 然后再 startup mount

此时数据库 PRODSTD 的 database_role 已经变成 physical standby

在 PROD 经行一次恢复

SQL>Alter database recover managed standby database disconnect from session

等后台恢复结束后

SQL>Alter database commit to switchover to primary

此时 PROD 的 database_role 已经变成 PRIMARY

运行脚本 q3.sql

SQL>@q3.sql

11. 在 PRODSTD 经行一次恢复

SQL>Alter database recover managed standby database disconnect from session;

等待后台结束即可