**第三周学习计划总结**

**ADG的搭建**

云和恩墨(北京)信息技术有限公司

技术顾问 燕鑫

http://www.enmotech.com

**文档控制：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序** | **版本号** | **更改人** | **日期** | **备注** |
| 1 | 1.0版 | 燕鑫 | 2018-04-27 | 初始版本 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编制** | 燕鑫 | （签字） | 日期 | 2018-04-27 |
| **校对** |  | （签字） | 日期 |  |
| **审核** |  | （签字） | 日期 |  |
| **批准** |  | （签字） | 日期 |  |

目录

[1. ADG的介绍 - 4 -](#_Toc513668031)

[2. 集群到集群的ADG搭建 - 4 -](#_Toc513668032)

[2.1 实验环境介绍： - 4 -](#_Toc513668033)

[2.1.1 主库实验环境 - 4 -](#_Toc513668034)

[2.1.2 备库实验环境 - 5 -](#_Toc513668035)

[2.2 主库配置 - 6 -](#_Toc513668036)

[2.2.1 归档，db\_unique\_name - 6 -](#_Toc513668037)

[2.2.2 force logging - 7 -](#_Toc513668038)

[2.2.3 配置主库静态监听 - 7 -](#_Toc513668039)

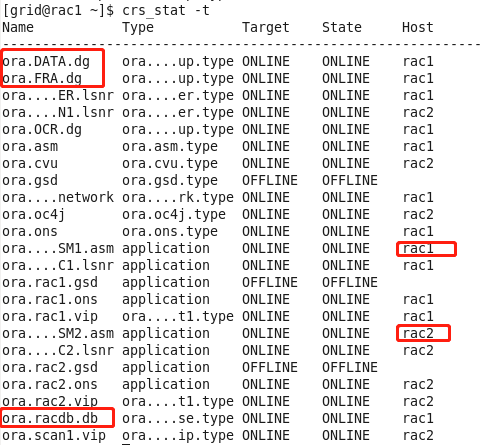
[2.2.4 配置主库tnsnames.ora - 9 -](#_Toc513668040)

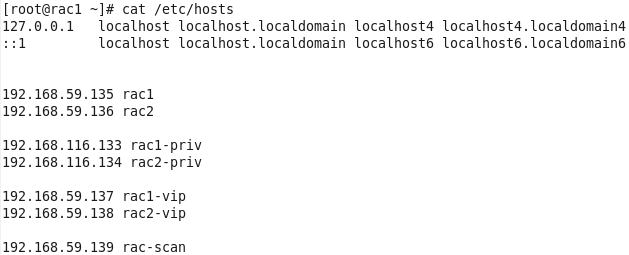
# ADG的介绍

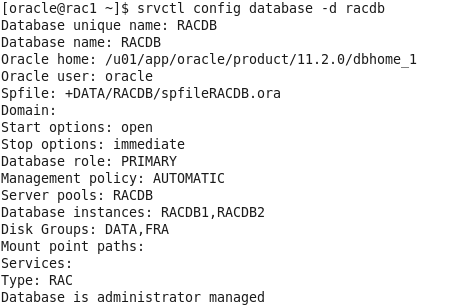
# 11g集群到集群的ADG搭建

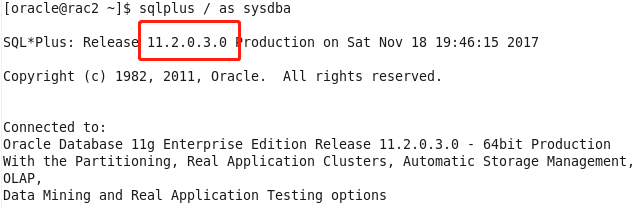
## 实验环境介绍：

### 主库实验环境







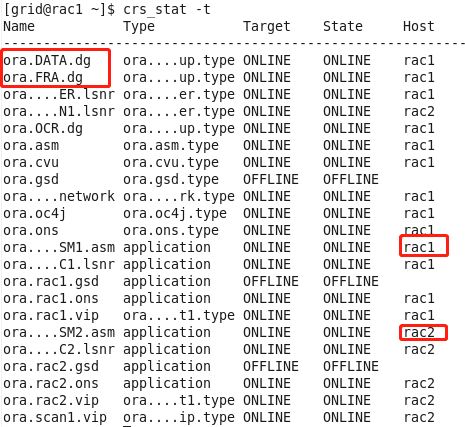


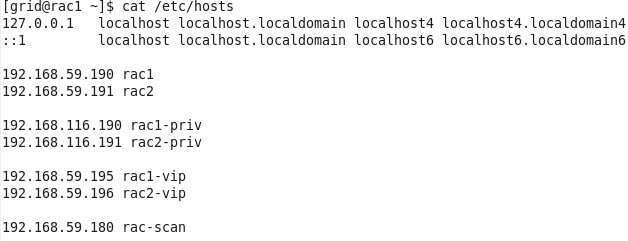


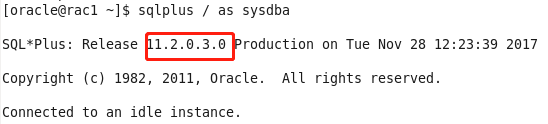


可以看到，主库是双节点集群，数据库版本是11.2.0.3.0，磁盘组有俩个+DATA,+FRA。

### 备库实验环境











看到备库也是双节点集群，并且是只有数据库软件，没有实例。磁盘组也有俩个+DATA,+FRA。

### 实验环境预期效果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Db\_NAME** | **DB\_TYPE** | DB\_ROLE | **Db\_uname\_name** | **监听端口** | **IP** | **TNS名字** |
| 主库 | racdb | RAC | PRIMARY | racdb | 1522 | 192.168.59.135/136 | racdb1522 |
| 备库 | racdb | RAC | PHYSICAL STANDBY | racdg | 1522 | 192.168.59.190/191 | racdg1522 |

## 主库配置

### 归档，db\_unique\_name

一般来说归档是开着的，没开必须开，开启归档需要将数据库启动到mount状态，然后alter database archivelog;

db\_unique\_name要改的话，也要重启数据库。

这里我们啥都不改，归档是已经打开着的。

SQL> show parameter db\_unique\_name;

NAME TYPE VALUE

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

db\_unique\_name string RACDB

SQL> archive log list

Database log mode Archive Mode

Automatic archival Enabled

Archive destination USE\_DB\_RECOVERY\_FILE\_DEST

Oldest online log sequence 129

Next log sequence to archive 130

Current log sequence 130

这里看到db\_unique\_name是RACDB，归档是开启着的。

### force logging

开启force logging是必须的，因为不管是啥DG，都是靠日志传输来保持主备一致的。

SQL> alter database no force logging;

Database altered.

SQL> alter database force logging;

Database altered.

SQL> select force\_logging from v$database;

FOR

---

YES

因为本身我主库是开启了force logging的，顺带演示下咋关。

### 配置主库静态监听

这里我们在grid用户用1522的端口来配置一个LSNR2的静态监听，用于主库备库的通信。

[grid@rac1 ~]$ srvctl add listener -l lsnr2 -p 1522

在rac1：

echo "

SID\_LIST\_LSNR2 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = RACDB)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/dbhome\_1)

(SID\_NAME = RACDB1)

)

)

DIAG\_ADR\_ENABLED\_LSNR2=OFF " >>/u01/app/11.2.0/grid/network/admin/listener.ora

ssh rac2

在rac2：

echo "

SID\_LIST\_LSNR2 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = RACDB)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/dbhome\_1)

(SID\_NAME = RACDB2)

)

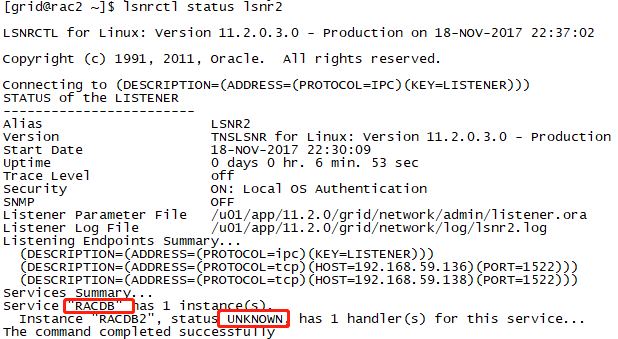
)

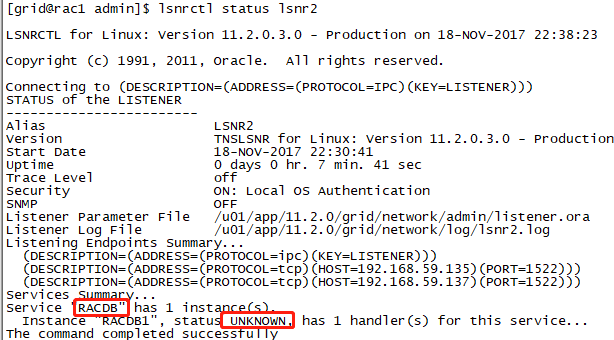
DIAG\_ADR\_ENABLED\_LSNR2=OFF " >>/u01/app/11.2.0/grid/network/admin/listener.ora

[grid@rac2 ~]$ srvctl start listener -l lsnr2

这里千万注意，是>>追加，而不是>

添加之后的效果如下图：





### 配置主库tnsnames.ora

我们在oracle用户配置tnsnames.ora。

[root@rac1 ~]# su - oracle

[oracle@rac1 ~]$

echo "

RACDB1522 =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.59.137)(PORT = 1522))

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.59.138)(PORT = 1522))

)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = RACDB)

)

)

RACDG1522 =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.59.195)(PORT = 1522))

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.59.196)(PORT = 1522))

)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = RACDG)

)

)">> /u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/tnsnames.ora

[oracle@rac1 ~]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/tnsnames.ora rac2:/u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/

### 主库修改参数

alter system set log\_archive\_config='dg\_config=(racdb,racdg)';

alter system set log\_archive\_dest\_1='location=use\_db\_recovery\_file\_dest valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=racdb';

alter system set log\_archive\_dest\_2='service=racdg1522 lgwr async valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=racdg';

alter system set log\_archive\_dest\_state\_2=enable;

alter system set fal\_server='racdg';

alter system set fal\_client='racdb';

alter system set standby\_file\_management=auto sid='\*';

alter system set log\_archive\_max\_processes=10 sid='\*';

alter system set db\_file\_name\_convert='racdg','racdb' scope=spfile sid='\*';

alter system set log\_file\_name\_convert='racdg','racdb' scope=spfile sid='\*';

这里注意一下，实验我们这里log\_archive\_dest\_1选择使用use\_db\_recovery\_file\_dest，如果主库是这样配置的，那备库就要做出相应的配置，也是use\_db\_recovery\_file\_dest。后面我们看。

还有着重注意一点：log\_archive\_dest\_2的service时tnsname！！！！

还有高亮部分是做switch over才需要配。

### 增加standby logfile

这里之所以要在主库添加standby logfile，是为了后续swich over用的。

添加standby logfile有俩个要求：

1. 每个线程比redo日志组多一组；
2. standby logfile日志组大小必须大于或等于redo日志组大小，官方建议一样大。

用如下查询查出每个线程有多少redo日志组，每个日志组有多大：

SQL> select count(group#),thread#,bytes/1024/1024 total\_size\_m from v$log group by thread#,bytes;

COUNT(GROUP#) THREAD# TOTAL\_SIZE\_M

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

2 1 50

2 2 50

我们看到有俩个线程（因为是双节点集群），每个线程有俩个redo日志组，每个redo日志组有50M。所以我们如下建立standby logfile：

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

SQL> select group#,thread#,bytes/1024/1024 total\_size\_m from v$standby\_log;

GROUP# THREAD# TOTAL\_SIZE\_M

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

5 1 50

6 1 50

7 1 50

8 2 50

9 2 50

10 2 50

可以看到，我们为每个线程分别建立了3个standby logfile日志组，每个大小50m。

### 全备以及做备库控制文件

echo "

rman target /<<EOF

run{

allocate channel ch01 type disk;

allocate channel ch02 type disk;

allocate channel ch03 type disk;

allocate channel ch04 type disk;

backup database format '/home/oracle/back/%d\_%U.FULL';

sql 'alter system archive log current';

backup current controlfile for standby format '/home/oracle/back/%d\_%U.CTL';

release channel ch01;

release channel ch02;

release channel ch03;

release channel ch04;

}

exit;

EOF

">backup.sh

source backup.sh

### 为备库准备参数文件

首先在一节点根据spfile生成pfile：

[oracle@rac1 back]$ sqlplus / as sysdba

SQL> create pfile='/home/oracle/back/initdg.ora' from spfile;

File created.

然后修改这个pfile，稍后用来作为备库启动的参数文件：

\*.\_\_db\_cache\_size=130023424

\*.\_\_java\_pool\_size=41943040

\*.\_\_large\_pool\_size=4194304

\*.\_\_pga\_aggregate\_target=230686720

\*.\_\_sga\_target=306184192

\*.\_\_shared\_io\_pool\_size=0

\*.\_\_shared\_pool\_size=163577856

\*.\_\_streams\_pool\_size=0

\*.archive\_lag\_target=0

\*.audit\_file\_dest='/u01/app/oracle/admin/RACDG/adump'

\*.audit\_trail='db'

\*.cluster\_database=true

\*.compatible='11.2.0.0.0'

\*.control\_files='+DATA/racdg/controlfile/current.260.960240097','+FRA/racdg/controlfile/current.256.960240097'

\*.db\_block\_size=8192

\*.db\_create\_file\_dest='+DATA'

\*.db\_domain=''

\*.db\_file\_name\_convert='racdb','racdg'

\*.db\_name='RACDB'

\*.db\_unique\_name='RACDG'

\*.db\_recovery\_file\_dest='+FRA'

\*.db\_recovery\_file\_dest\_size=4294967296

\*.dg\_broker\_start=FALSE

\*.diagnostic\_dest='/u01/app/oracle'

\*.dispatchers='(PROTOCOL=TCP) (SERVICE=RACDBXDB)'

\*.fal\_client='racdg'

\*.fal\_server='racdb'

RACDG1.instance\_number=1

RACDG2.instance\_number=2

\*.log\_archive\_config='dg\_config=(racdb,racdg)'

\*.log\_archive\_dest\_1='location=use\_db\_recovery\_file\_dest valid\_for=(all\_logfiles,all\_roles) db\_unique\_name=racdg'

\*.log\_archive\_dest\_2='service=racdb lgwr async valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=racdb'

\*.log\_archive\_dest\_state\_2='ENABLE'

\*.log\_archive\_format='%t\_%s\_%r.arc'

\*.log\_archive\_max\_processes=10

\*.log\_archive\_min\_succeed\_dest=1

\*.log\_archive\_trace=0

\*.log\_file\_name\_convert='racdb','racdg'

\*.memory\_target=536870912

\*.open\_cursors=300

\*.processes=150

\*.remote\_listener='rac-scan:1521'

\*.remote\_login\_passwordfile='exclusive'

\*.standby\_file\_management='AUTO'

RACDG2.thread=2

RACDG1.thread=1

RACDG2.undo\_tablespace='UNDOTBS2'

RACDG1.undo\_tablespace='UNDOTBS1'

#### 参数文件修改总结

总结一下：

1. 高亮的部分是需要注意的地方
2. 添加\*.db\_unique\_name=xx
3. 必须要分别指定的是instance\_number, thread, undo\_tablespace(这个其实无所谓)
4. 因为备库是集群，所以必须\*.cluster\_database=true
5. 备库的\*.log\_archive\_config必须和主库保持一致，并且必须有，否则在应用日志的时候会报错，稍后模拟。
6. \*.standby\_file\_management=’auto’这个必须指定，否则主库添加数据文件，备库要手动添加相应的数据文件。
7. 除了上面的，其余的把之前的实例名都改成\*，重复的就dd掉。
8. 备库的db\_name不能改，实例名可以修改，因为db\_name同时也记录再控制文件中。

### 拷贝备份文件、密码文件、修改好的参数文件到备库

为了尽可能多的做一些实验，我们这里在备库建一个与主库备份时指定的不一样名称的文件夹：/home/oracle/backup(主库的是/home/oracle/back)

拷贝备份文件和参数文件到备库一节点：

[oracle@rac1 back]$ scp /home/oracle/back/\* 192.168.59.190:/home/oracle/backup/

拷贝参数文件到备库二节点：

[oracle@rac1 back]$ scp /home/oracle/back/initdg.ora 192.168.59.191:/home/oracle/backup/

拷贝主库一节点的密码文件到备库一、二节点：

[oracle@rac1 back]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/dbs/orapwRACDB1 192.168.59.190:/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/orapwRACDG1

[oracle@rac1 back]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/dbs/orapwRACDB1 192.168.59.191:/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/orapwRACDG2

## 备库配置

### 修改备库oracle用户的.bash\_profile，配置环境变量

备库一节点的配置如下：

export ORACLE\_SID=RACDG1

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0/dbhome\_1

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib

export NLS\_DATE\_FORMAT="yyyy-mm-dd HH24:MI:SS"

export TMP=/tmp

export TMPDIR=$TMP

export PATH=$PATH:$ORACLE\_HOME/bin

备库二节点配置如下：

export ORACLE\_SID=RACDG2

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0/dbhome\_1

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib

export NLS\_DATE\_FORMAT="yyyy-mm-dd HH24:MI:SS"

export TMP=/tmp

export TMPDIR=$TMP

export PATH=$PATH:$ORACLE\_HOME/bin

### 配置备库静态监听

同样我们在grid用户用1522的端口来配置一个LSNR2的静态监听，用于主库备库的通信。

[grid@rac1 ~]$ srvctl add listener -l lsnr2 -p 1522

在rac1：

echo "

SID\_LIST\_LSNR2 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = RACDG)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/dbhome\_1)

(SID\_NAME = RACDG1)

)

)

DIAG\_ADR\_ENABLED\_LSNR2=OFF " >>/u01/app/11.2.0/grid/network/admin/listener.ora

ssh rac2

在rac2：

echo "

SID\_LIST\_LSNR2 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = RACDG)

(ORACLE\_HOME = /u01/app/oracle/product/11.2.0/dbhome\_1)

(SID\_NAME = RACDG2)

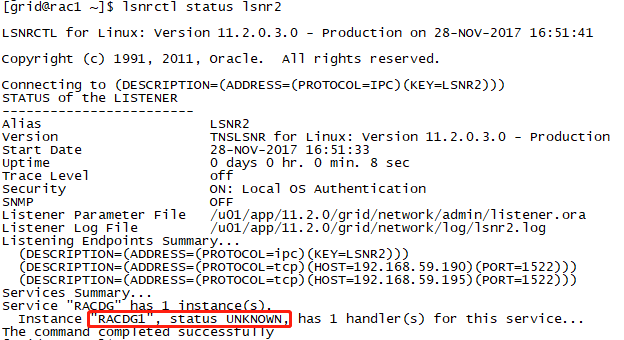
)

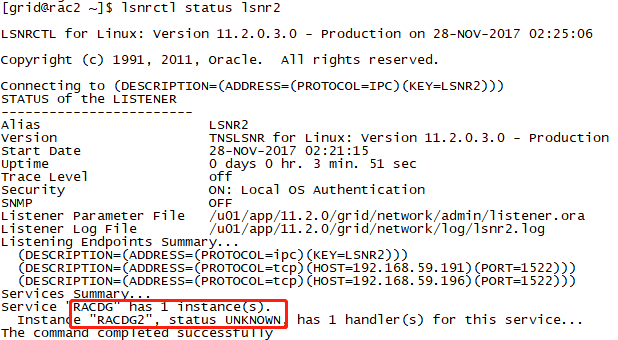
)

DIAG\_ADR\_ENABLED\_LSNR2=OFF " >>/u01/app/11.2.0/grid/network/admin/listener.ora

[grid@rac2 ~]$ srvctl start listener -l lsnr2

添加之后的效果如下图：





### 拷贝主库的tnsnames.ora到备库

这里就不配置备库的tnsnames.ora了，直接将主库的拷贝过来即可：

[oracle@rac1 back]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/tnsnames.ora 192.168.59.190:/u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/

[oracle@rac1 back]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/tnsnames.ora 192.168.59.191:/u01/app/oracle/product/11.2.0/dbhome\_1/network/admin/

### 创建参数文件中相关的目录（adump）



这个目录备库俩个节点都要要建：

[oracle@rac1 ~]$ mkdir -p /u01/app/oracle/admin/RACDG/adump

[oracle@rac2 ~]$ mkdir -p /u01/app/oracle/admin/RACDG/adump

### 启动备库节点一到nomount并生成spfile

这里只启动一节点，在dg搭建完成后将第二个节点加入即可。

[oracle@rac1 ~]$ sqlplus / as sysdba

SQL> startup nomount pfile='/home/oracle/backup/initdg.ora';

ORACLE instance started.

Total System Global Area 534462464 bytes

Fixed Size 2230072 bytes

Variable Size 398461128 bytes

Database Buffers 130023424 bytes

Redo Buffers 3747840 bytes

然后创建spfile：

SQL> create spfile='+data' from pfile='/home/oracle/backup/initdg.ora';

高亮部分必须指定！！！也就是说pfile的绝对路径必须指定，不能写成create spfile from pfile;这个报错后面总结。这个只针对RAC。

编辑备库一节点的pfile：

vi $ORACLE\_HOME/dbs/initRACDG1.ora

spfile='+data/RACDG/PARAMETERFILE/spfile.257.961264711'

这里要是建spfile没指定明确的路径和名字，就到asmcmd里面查一下，要是指定了，就直接把指定的路径名字填写到pfile里面就行。RAC才需要这么做。

### 恢复备库备份文件并启动备库到mount

先恢复控制文件。

RMAN> restore standby controlfile from '/home/oracle/backup/RACDB\_1msjtqmb\_1\_1.CTL';

Starting restore at 2017-11-28 18:35:29

using target database control file instead of recovery catalog

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=35 instance=RACDG1 device type=DISK

channel ORA\_DISK\_1: restoring control file

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:01

output file name=+DATA/racdg/controlfile/current.280.961266395

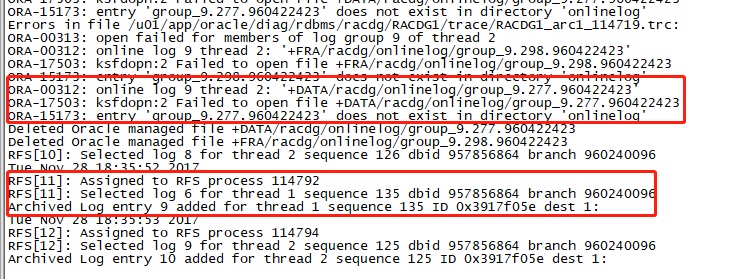
output file name=+FRA/racdg/controlfile/current.261.961266395

Finished restore at 2017-11-28 18:35:30

然后将数据库启动到mount，如果前面参数文件中修改了dbname，这里就会报错（ORA-01103: database name 'RACDB' in control file is not 'RACDG'）：

RMAN> alter database mount;

这时我们观察日志：



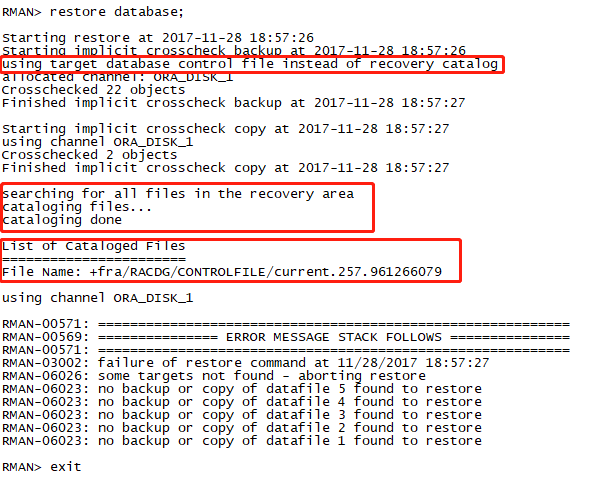
可以看到日志已经在往备库传了。但是有报错，因为没办法应用日志。

### 恢复备份数据库

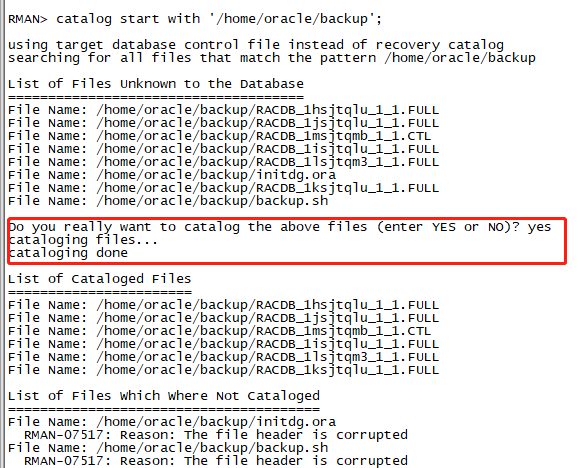
前面我们修改了路径，就是为了实验如下这条命令：

RMAN> catalog start with '/home/oracle/backup';

如果我们直接restore database,会报如下错误：



因为备份文件的路径会记录在控制文件中，路径改变了，就找不到了，所以我们用上述的命令可以重新指定这个路径，这条命令必须在controlfile恢复后，数据库启动到mount才能执行。



然后执行restore database：

RMAN> restore database;

### 增加备库的standby logfile

前面给主库加，是为了给后续switch over做准备，现在在备库加就是必须的，原则还是一样：

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 1 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

alter database add standby logfile thread 2 size 50m;

SQL> select group#,thread#,bytes/1024/1024 total\_size\_m from v$standby\_log;

GROUP# THREAD# TOTAL\_SIZE\_M

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

5 1 50

6 1 50

7 1 50

8 2 50

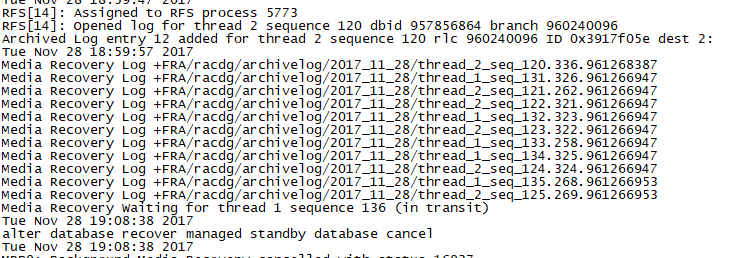
9 2 50

10 2 50

### 备库应用日志

SQL> alter database recover managed standby database disconnect ;

可以看到：



当看到日志都追上来了以后，就可以进行下一步，启动备库到adg。

### 启动备库到ADG

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

### 验证ADG

现在主库进行建表插入的操作：

SQL> conn scott/tiger

Connected.

SQL> create table yxikk(id number);

Table created.

SQL> insert into yxikk values(1);

1 row created.

SQL> commit;

Commit complete.

立马去备库一节点进行查询：

SQL> select \* from scott.yxikk;

ID

----------

1

### 添加备库二节点到GRID

这里先把备库一节点的参数文件拷贝过来：

[oracle@rac1 ~]$ scp /u01/app/oracle/product/11.2.0/dbhome\_1/dbs/initRACDG1.ora rac2:/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/initRACDG2.ora

然后直接启动就好了：

[oracle@rac2 dbs]$ sqlplus / as sysdba

SQL> startup

ORACLE instance started.

Total System Global Area 534462464 bytes

Fixed Size 2230072 bytes

Variable Size 398461128 bytes

Database Buffers 130023424 bytes

Redo Buffers 3747840 bytes

Database mounted.

Database opened.

SQL> select open\_mode from v$database;

OPEN\_MODE

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

READ ONLY WITH APPLY

然后执行如下操作：

[oracle@rac2 dbs]$ srvctl add database -d racdg -o $ORACLE\_HOME -c RAC -p +data/RACDG/PARAMETERFILE/spfile.257.961264711 -r physical\_standby

[oracle@rac2 dbs]$ srvctl add instance -d racdg -i RACDG1 -n rac1

[oracle@rac2 dbs]$ srvctl add instance -d racdg -i RACDG2 -n rac2

[oracle@rac2 dbs]$ srvctl start instance -d racdg -i RACDG1

[oracle@rac2 dbs]$ srvctl start instance -d racdg -i RACDG2

注意这是oracle用户下执行的。同时，这里注意！实例名区分大小写！！！

用srvctl启动数据库实例，系统会自己在oracle用户的$ORACLE\_HOME/dbs下生成相应的pfile。因为我们在上面添加database的时候指定了相应的spfile。

[oracle@rac1 dbs]$ ls

orapwRACDG1

[oracle@rac2 dbs]$ ls

orapwRACDG2

看到这里删的就剩密码文件了，然后我们用srvctl启动数据库：

[oracle@rac2 dbs]$ srvctl start database -d racdg

[oracle@rac2 dbs]$ ls

hc\_RACDG2.dat initRACDG2.ora orapwRACDG2

[oracle@rac2 dbs]$ cat initRACDG2.ora

SPFILE='+DATA/RACDG/PARAMETERFILE/spfile.257.961264711' # line added by Agent

启动以后数据库只会是read only，所以我们这里还要进行一步应用日志：

SQL> alter database recover managed standby database using current logfile disconnect;

## 实验

### 添加数据文件

先检查主库的数据文件：

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdb/datafile/users.259.960240045

UNDOTBS1 +DATA/racdb/datafile/undotbs1.258.960240045

SYSAUX +DATA/racdb/datafile/sysaux.257.960240045

SYSTEM +DATA/racdb/datafile/system.256.960240045

EXAMPLE +DATA/racdb/datafile/example.264.960240101

UNDOTBS2 +DATA/racdb/datafile/undotbs2.265.960240197

再看备库的：

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdg/datafile/users.270.961268315

UNDOTBS1 +DATA/racdg/datafile/undotbs1.264.961268305

SYSAUX +DATA/racdg/datafile/sysaux.267.961268309

SYSTEM +DATA/racdg/datafile/system.269.961268315

EXAMPLE +DATA/racdg/datafile/example.263.961268305

UNDOTBS2 +DATA/racdg/datafile/undotbs2.268.961268309

现在主库新建个表空间：

SQL> create tablespace yx datafile size 10m autoextend on;

Tablespace created.

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdb/datafile/users.259.960240045

UNDOTBS1 +DATA/racdb/datafile/undotbs1.258.960240045

SYSAUX +DATA/racdb/datafile/sysaux.257.960240045

SYSTEM +DATA/racdb/datafile/system.256.960240045

EXAMPLE +DATA/racdb/datafile/example.264.960240101

UNDOTBS2 +DATA/racdb/datafile/undotbs2.265.960240197

YX +DATA/racdb/datafile/yx.272.960452937

现在来看备库：

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdg/datafile/users.270.961268315

UNDOTBS1 +DATA/racdg/datafile/undotbs1.264.961268305

SYSAUX +DATA/racdg/datafile/sysaux.267.961268309

SYSTEM +DATA/racdg/datafile/system.269.961268315

EXAMPLE +DATA/racdg/datafile/example.263.961268305

UNDOTBS2 +DATA/racdg/datafile/undotbs2.268.961268309

YX +DATA/racdg/datafile/yx.287.961227739

假如我们这里把备库的standby\_file\_management置为MANUAL，那顾名思义就是手动管理：

SQL> alter system set standby\_file\_management=MANUAL sid='\*';

System altered.

现在我们再到主库给YX表空间添加数据文件：

SQL> alter tablespace yx add datafile size 10m;

Tablespace altered.

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdb/datafile/users.259.960240045

UNDOTBS1 +DATA/racdb/datafile/undotbs1.258.960240045

SYSAUX +DATA/racdb/datafile/sysaux.257.960240045

SYSTEM +DATA/racdb/datafile/system.256.960240045

EXAMPLE +DATA/racdb/datafile/example.264.960240101

UNDOTBS2 +DATA/racdb/datafile/undotbs2.265.960240197

YX +DATA/racdb/datafile/yx.272.960452937

YX +DATA/racdb/datafile/yx.273.960453721

看备库：

SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdg/datafile/users.270.961268315

UNDOTBS1 +DATA/racdg/datafile/undotbs1.264.961268305

SYSAUX +DATA/racdg/datafile/sysaux.267.961268309

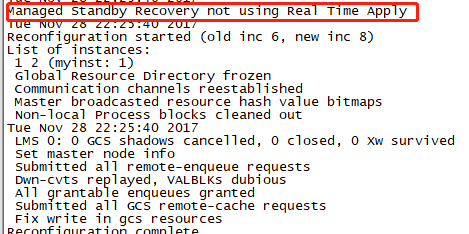
SYSTEM +DATA/racdg/datafile/system.269.961268315

EXAMPLE +DATA/racdg/datafile/example.263.961268305

UNDOTBS2 +DATA/racdg/datafile/undotbs2.268.961268309

YX +DATA/racdg/datafile/yx.287.961227739

备库并无变化，但日志是有变化的：



备库自己把实时应用取消了：

SQL> select open\_mode from v$database;

OPEN\_MODE

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

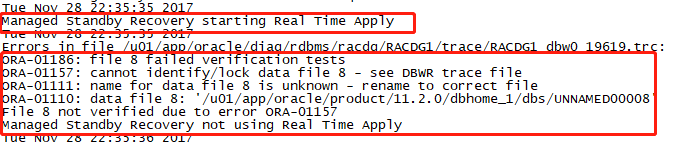
READ ONLY

果然，数据变回了read only。那现在我们来把备库变为ADG。

SQL> alter database recover managed standby database using current logfile disconnect;

Database altered.

然后看日志：



数据库拒绝启动到ADG，那file 8是谁？！

SQL> select file#,name from v$datafile where file#=8;

FILE # NAME

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

8 +DATA/racdb/datafile/yx.273.960453721

正是我们刚才新加的数据文件，我们这里不手动加这个文件，我们把参数改回来并再次开启日志实时应用看看：

SQL> alter system set standby\_file\_management=auto sid='\*';

System altered.

SQL> select open\_mode from v$database;

OPEN\_MODE

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

READ ONLY

SQL> alter database recover managed standby database using current logfile disconnect;

Database altered.

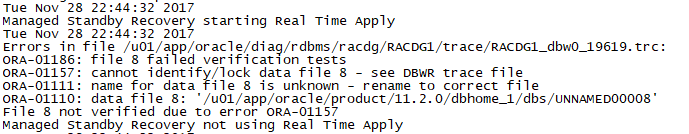
SQL> show parameter standby\_file\_man

NAME TYPE VALUE

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

standby\_file\_management string AUTO

一通操作下来看日志，发现没卵用：



看来只能手动添加了：

SQL> alter tablespace yx add datafile size 10m;

ERROR at line 1:

ORA-16000: database open for read-only access

看，这样做是不对的。看上面的日志报错，和在上面的日志报错的区别，这里使standby\_file\_management变为auto后，数据文件是生成了的，但是不能自动加入到数据库中，我们现在直接添加：

SQL> alter database create datafile '/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/UNNAMED00008' as '+DATA/racdb/datafile/yx.273.960453721';

ERROR at line 1:

ORA-01276: Cannot add file +DATA/racdb/datafile/yx.273.960453721. File has an Oracle Managed Files file name.

SQL> alter database create datafile '/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/UNNAMED00008' as '+data';

ERROR at line 1:

ORA-01275: Operation CREATE DATAFILE is not allowed if standby file management is automatic.

发现俩个问题：

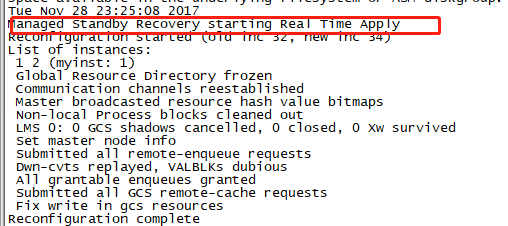
1. 如果备库数据文件是OMF管理的，也就是说db\_create\_file\_dest有值，那就不能这样指定名字添加。本来OMF管理，主备库的数据文件名字也是不一样的。
2. 要添加必须是standby\_file\_management为manual时才行。

所以我们做如下操作：

SQL> alter system set standby\_file\_management=manual sid='\*';

SQL> alter database create datafile '/u01/app/oracle/product/11.2.0/dbhome\_1/dbs/UNNAMED00008' as '+data';

数据文件被成功添加了。让后我们重新把standby\_file\_management改为auto，并且启动到ADG：



SQL> select tablespace\_name,file\_name from dba\_data\_files;

TABLESPACE\_NAME FILE\_NAME

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

USERS +DATA/racdg/datafile/users.270.961268315

UNDOTBS1 +DATA/racdg/datafile/undotbs1.264.961268305

SYSAUX +DATA/racdg/datafile/sysaux.267.961268309

SYSTEM +DATA/racdg/datafile/system.269.961268315

EXAMPLE +DATA/racdg/datafile/example.263.961268305

UNDOTBS2 +DATA/racdg/datafile/undotbs2.268.961268309

YX +DATA/racdg/datafile/yx.287.961227739

YX +DATA/racdg/datafile/yx.286.961231193

发现日志显示实施应用开启了，并且数据文件也成功恢复上去了。

### 主库备库switch over

#### 把主库和备库的2节点停掉

只留一个实例！

#### 确认主库没有日志gap

SQL> select STATUS, GAP\_STATUS from V$ARCHIVE\_DEST\_STATUS where DEST\_ID = 2;

STATUS GAP\_STATUS

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

VALID NO GAP

#### 确认所有重做日志已全部应用到备库

SQL> select NAME, VALUE, DATUM\_TIME from V$DATAGUARD\_STATS;

NAME VALUE DATUM\_TIME

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

transport lag +00 00:00:00 11/29/2017 01:12:38

apply lag +00 00:00:00 11/29/2017 01:12:38

apply finish time +00 00:00:00.000

estimated startup time 20

其中estimated startup time是物理备库启动所需要的估计时间，重要的是datum\_time，如果这个值不变化了，说明此时主库备库之间的通信出了问题，肯定也就有了gap。

#### 主库查看当前数据库状态

SQL> select database\_role,switchover\_status from v$database;

DATABASE\_ROLE SWITCHOVER\_STATUS

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

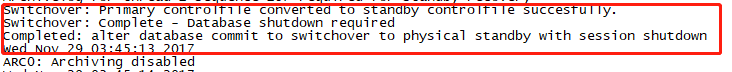
PRIMARY TO STANDBY

如果时to standby或者session active，说明主库可以进行切换。

#### 进行主库到备库的switch over

SQL> alter database commit to switchover to physical standby with session shutdown;

Database altered.



#### 确认备库是否可以切换

SQL> select SWITCHOVER\_STATUS from V$DATABASE;

SWITCHOVER\_STATUS

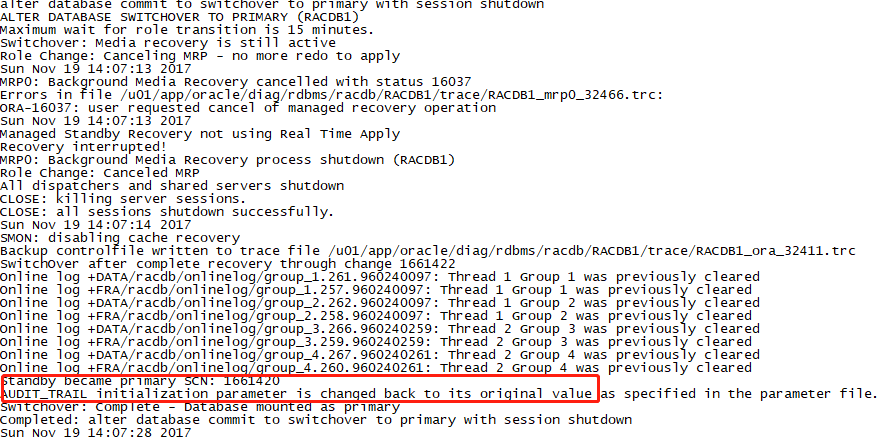
--------------------

TO PRIMARY

#### 进行备库到主库的switch over

SQL> alter database commit to switchover to primary with session shutdown;

Database altered.



SQL> select open\_mode from v$database;

OPEN\_MODE

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

MOUNTED

SQL> alter database open;

Database altered.

#### 启动备库到ADG

SQL> startup mount

ORACLE instance started.

Database mounted.

SQL> alter database open;

Database altered.

SQL> recover managed standby database using current logfile disconnect;

Media recovery complete.

SQL> select open\_mode from v$database;

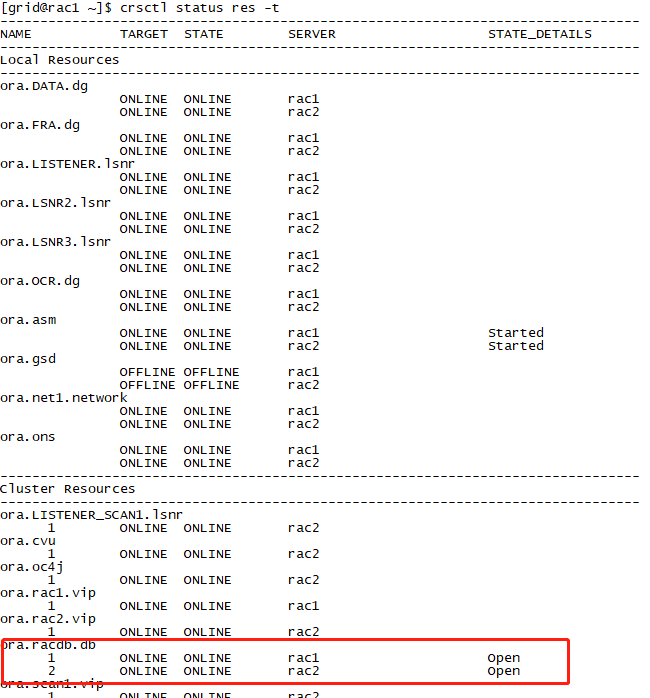
OPEN\_MODE

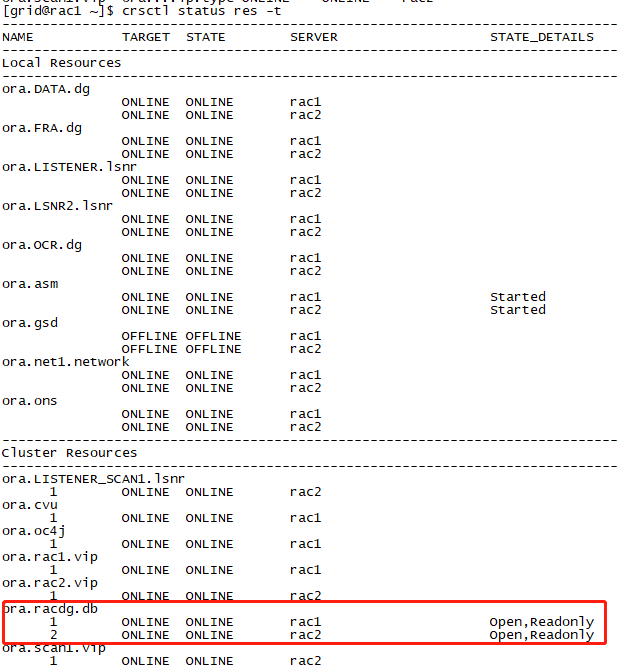
--------------------

READ ONLY WITH APPLY

这里一定要先启动到mount阶段，再启动到open阶段，否则回报system数据文件需要恢复的错误。

#### 把主库和备库的2节点重新启动





# 12c单实例到单实例的ADG搭建

## 实验环境介绍：

### 主库实验环境

[oracle@host01 ~]$ echo $ORACLE\_HOME

/u01/app/oracle/product/12c/db\_1

[oracle@host01 ~]$ sqlplus -v

SQL\*Plus: Release 12.2.0.1.0 Production

SQL> show parameter instance\_name

NAME TYPE VALUE

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

instance\_name string PRODCDB

### 备库实验环境

[oracle@host02 ~]$ echo $ORACLE\_HOME

/u01/app/oracle/product/12c/db\_1

[oracle@host02 ~]$ sqlplus -v

SQL\*Plus: Release 12.2.0.1.0 Production

### 实验环境预期效果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Db\_NAME** | **DB\_TYPE** | DB\_ROLE | **Db\_uname\_name** | **监听端口** | **IP** | **TNS名字** |
| 主库 | PRODCDB | 单实例 | PRIMARY | PRODCDB | 1522 | 192.168.12.10 | prodcdb1522 |
| 备库 | PRODCDB | 单实例 | PHYSICAL STANDBY | SBDB | 1522 | 192.168.12.20 | sbdb1522 |

## 主库配置

### 归档，db\_unique\_name

SQL> show parameter db\_unique\_name

NAME TYPE VALUE

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

db\_unique\_name string PRODCDB

SQL> archive log list

Database log mode Archive Mode

Automatic archival Enabled

Archive destination /home/oracle/arch

Oldest online log sequence 53

Next log sequence to archive 55

Current log sequence 55

这里看到db\_unique\_name是PRODCDB，归档是开启着的。

### force logging

开启force logging:

SQL> alter database force logging;

Database altered.

SQL> select force\_logging from v$database;

FOR

---

YES

### 配置主库静态监听

这里我们在用1522的端口来配置一个LSNR1522的静态监听，用于主库备库的通信:

[oracle@host01 ~]$

echo "

LSNR1522 =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = host01)(PORT = 1522))

)

SID\_LIST\_LSNR1522 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = PRODCDB)

(ORACLE\_HOME = /u01/app/oracle/product/12c/db\_1)

(SID\_NAME = PRODCDB)

)

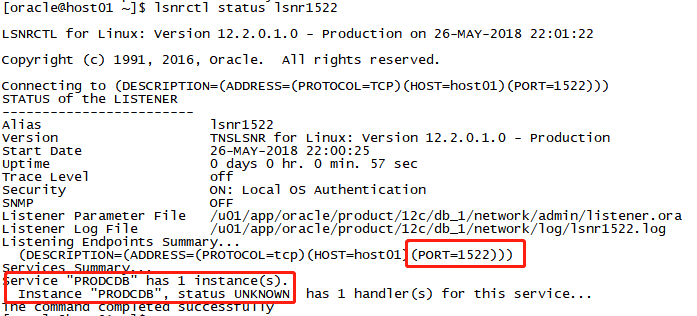
)

DIAG\_ADR\_ENABLED\_LSNR1522=OFF " >>/u01/app/oracle/product/12c/db\_1/network/admin/listener.ora

[oracle@host01 ~]$ lsnrctl start lsnr1522

这里千万注意，是>>追加，而不是>

添加之后的效果如下图：



### 配置主库tnsnames.ora

我们在oracle用户配置tnsnames.ora。

[oracle@host01 ~]$

echo "

PRODCDB1522 =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.12.10)(PORT = 1522))

)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = PRODCDB)

)

)

SBDB1522 =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.12.20)(PORT = 1522))

)

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = SBDB)

)

)">> /u01/app/oracle/product/12c/db\_1/network/admin/tnsnames.ora

### 主库修改参数

alter system set log\_archive\_config='dg\_config=(prodcdb,sbdb)';

alter system set log\_archive\_dest\_2='service=sbdb1522 lgwr async valid\_for=(online\_logfiles,primary\_role) db\_unique\_name=sbdb';

alter system set log\_archive\_dest\_state\_2=enable;

alter system set fal\_server='sbdb';

alter system set fal\_client='prodcdb';

alter system set standby\_file\_management=auto;

alter system set log\_archive\_max\_processes=10;

alter system set db\_file\_name\_convert='PRODCDB','SBDB' scope=spfile;

alter system set log\_file\_name\_convert='PRODCDB','SBDB' scope=spfile;

高亮部分是做switch over才需要配。

### 增加standby logfile

SQL> select count(group#),thread#,bytes/1024/1024 total\_size\_m from v$log group by thread#,bytes;

COUNT(GROUP#) THREAD# TOTAL\_SIZE\_M

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

3 1 200

alter database add standby logfile size 200m;

alter database add standby logfile size 200m;

alter database add standby logfile size 200m;

alter database add standby logfile size 200m;

SQL> select group#,thread#,bytes/1024/1024 total\_size\_m from v$standby\_log;

GROUP# THREAD# TOTAL\_SIZE\_M

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

4 0 200

5 0 200

6 0 200

7 0 200

### 密码文件到备库

拷贝主库的密码文件到备库：

[oracle@host01 ~]$ scp /u01/app/oracle/product/12c/db\_1/dbs/orapwPRODCDB host02:/u01/app/oracle/product/12c/db\_1/dbs/orapwSBDB

## 备库配置

### 修改备库oracle用户的.bash\_profile，配置环境变量

备库的配置如下：

[oracle@host02 ~]$ cat .bash\_profile

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

# User specific environment and startup programs

export ORACLE\_SID=SBDB

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=/u01/app/oracle/product/12c/db\_1

export PATH=$ORACLE\_HOME/bin:/usr/sbin:$PATH

### 配置备库静态监听

同样我们用1522的端口来配置一个LSNR1522的静态监听，用于主库备库的通信：

[oracle@host02 ~]$

echo "

LSNR1522 =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = host02)(PORT = 1522))

)

SID\_LIST\_LSNR1522 =

(SID\_LIST =

(SID\_DESC =

(GLOBAL\_DBNAME = SBDB)

(ORACLE\_HOME = /u01/app/oracle/product/12c/db\_1)

(SID\_NAME = SBDB)

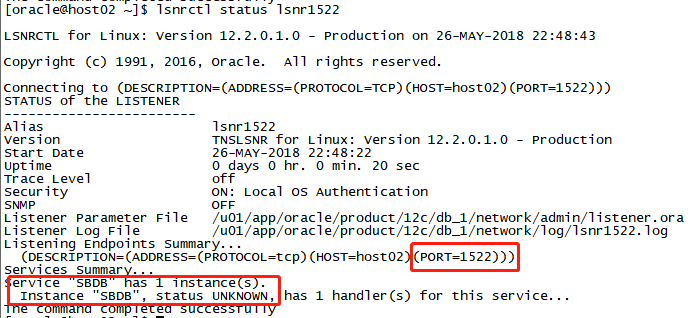
)

)

DIAG\_ADR\_ENABLED\_LSNR1522=OFF " >>/u01/app/oracle/product/12c/db\_1/network/admin/listener.ora

[oracle@host02 ~]$ lsnrctl start lsnr1522

添加之后的效果如下图：



### 拷贝主库的tnsnames.ora到备库

这里就不配置备库的tnsnames.ora了，直接将主库的拷贝过来即可：

[oracle@host01 ~]$ scp /u01/app/oracle/product/12c/db\_1/network/admin/tnsnames.ora host02:/u01/app/oracle/product/12c/db\_1/network/admin/tnsnames.ora

### 创建参数文件中相关的目录（adump）

[oracle@host02 ~]$ mkdir -p /u01/app/oracle/admin/SBDB/adump

### 创建pfile并启动备库到nomount

创建pfile：

[oracle@host02 ~]$ echo "DB\_NAME=PRODCDB">>$ORACLE\_HOME/dbs/initSBDB.ora

启动备库到nomount：

[oracle@host02 ]$ sqlplus / as sysdba

SQL\*Plus: Release 12.2.0.1.0 Production on Sat May 26 23:31:06 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup nomount

ORACLE instance started.

Total System Global Area 394264576 bytes

Fixed Size 8620944 bytes

Variable Size 327156848 bytes

Database Buffers 50331648 bytes

Redo Buffers 8155136 bytes

### duplicate建备库

[oracle@host02 ~]$ rman target sys/oracle@prodcdb1522 AUXILIARY sys/oracle@sbdb1522

Recovery Manager: Release 12.2.0.1.0 - Production on Sat May 26 23:18:29 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

connected to target database: PRODCDB (DBID=2962887328)

connected to auxiliary database: PRODCDB (not mounted)

run

{

allocate channel c1 type disk;

allocate channel c2 type disk;

allocate channel c3 type disk;

allocate channel c4 type disk;

allocate auxiliary channel a1 type disk;

allocate auxiliary channel a2 type disk;

allocate auxiliary channel a3 type disk;

allocate auxiliary channel a4 type disk;

duplicate target database for standby from active database nofilenamecheck

spfile

parameter\_value\_convert 'PRODCDB','SBDB'

set db\_name='PRODCDB'

set db\_unique\_name='SBDB'

set diagnostic\_dest=' /u01/app/oracle'

set db\_recovery\_file\_dest='/home/oracle/flash'

set db\_recovery\_file\_dest\_size='4G'

set fal\_client='SBDB'

set fal\_server='PRODCDB'

set log\_archive\_dest\_1='location=/home/oracle/arch'

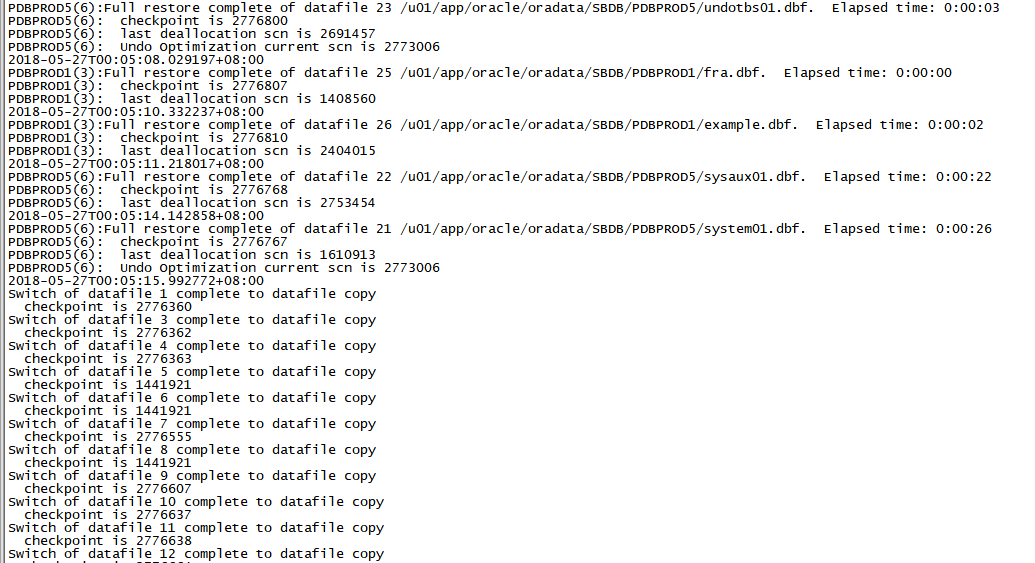
set log\_archive\_dest\_2='service=prodcdb1522 async lgwr valid\_for=(online\_logfile,primary\_role) db\_unique\_name=prodcdb'

set standby\_file\_management='AUTO';

}

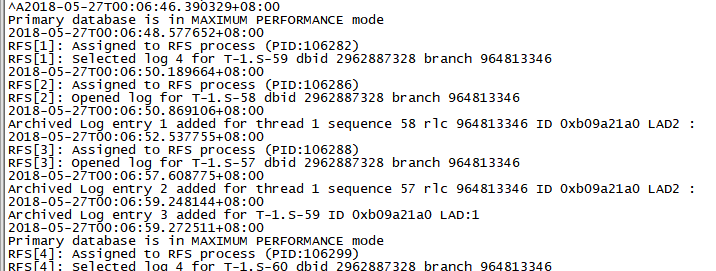
db\_name在这样duplicate的时候一定要加进来，不要以为init.ora里有了就不用显示指定了，如果不指定，oracle会把db\_name改成db\_unique\_name。

下面是duplicate过程中，备库的日志信息：



### 增加备库的standby logfile

我们看到备库alert日志中已经有如下的显示了：



这里本应该添加standby log的，但是因为事先在主库加了standby log，所以duplicate的时候把备库的standby log也建好了。

SQL> select group#,thread#,bytes/1024/1024 from v$standby\_log;

GROUP# THREAD# BYTES/1024/1024

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

4 1 200

5 0 200

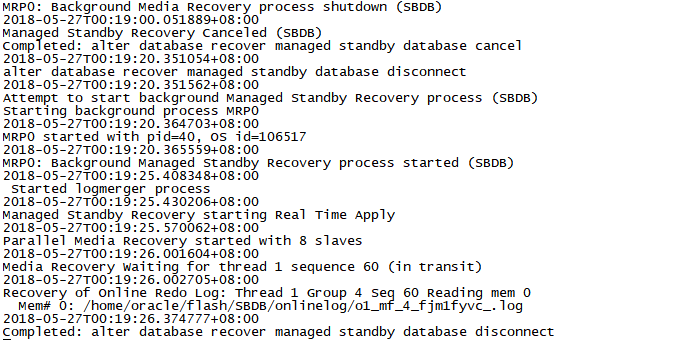
6 0 200

7 0 200

### 备库应用日志

SQL> alter database recover managed standby database disconnect ;

可以看到：



当看到日志都追上来了以后，就可以进行下一步，启动备库到adg。

### 启动备库到ADG

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

SQL> alter pluggable database all open;

Pluggable database altered.

SQL> show pdbs

CON\_ID CON\_NAME OPEN MODE RESTRICTED

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

2 PDB$SEED READ ONLY NO

3 PDBPROD1 READ ONLY NO

4 PDBPROD2 READ ONLY NO

5 PDBPROD3 READ ONLY NO

6 PDBPROD READ ONLY NO

### 验证ADG

现在主库进行建表插入的操作：

SQL> conn scott/tiger@pdbprod1

Connected.

SQL> create table yyy(id number);

Table created.

SQL> insert into yyy values(1);

1 row created.

SQL> commit;

Commit complete.

立马去备库进行查询：

SQL> conn scott/tiger@pdbprod1

Connected.

SQL> select \* from yyy;

ID

----------

1

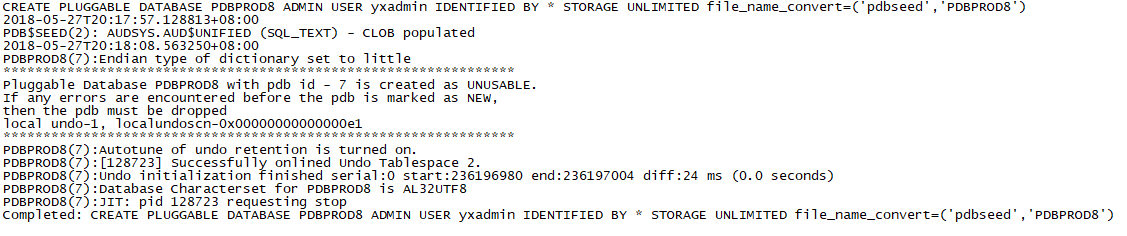
### 实验

#### 主库添加pdb

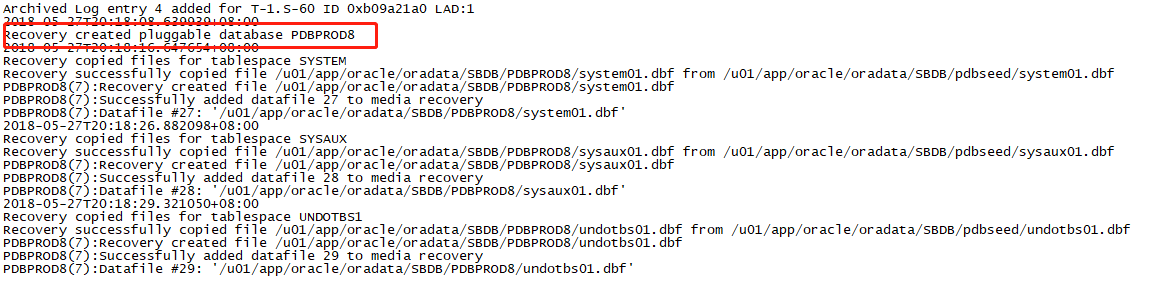
首先在主库添加pdb：

SQL> CREATE PLUGGABLE DATABASE PDBPROD8 ADMIN USER yxadmin IDENTIFIED BY oracle STORAGE UNLIMITED file\_name\_convert=('pdbseed','PDBPROD8');

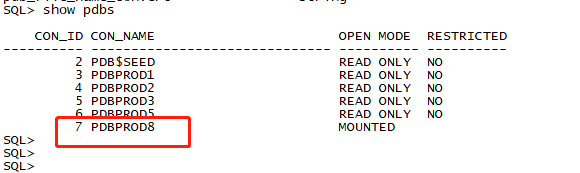
然后查看主库日志：



之后在备库查看日志：

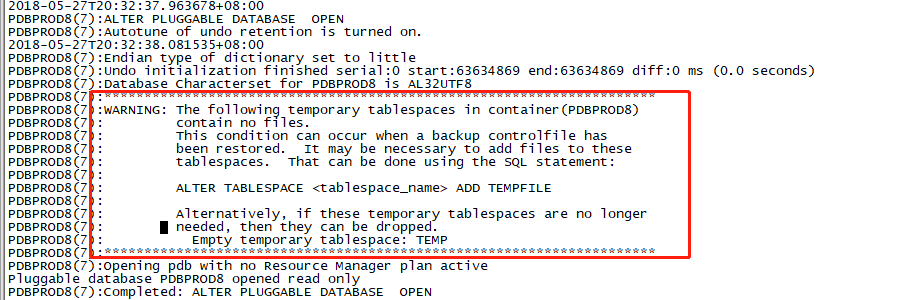


看到已经在备库进行了恢复，然后到备库show pdbs



看到确实有了。

把备库pdb起起来看日志：



这是因为DG不会处理临时表空间的，我们看主库，在刚建的pdbprod8下查看：

SQL> col file\_name for a80

SQL> select file\_name,tablespace\_name,status from dba\_temp\_files;

FILE\_NAME TABLESPACE\_NAME STATUS

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

/u01/app/oracle/oradata/PRODCDB/PDBPROD8/temp012018-01-07\_19-42-49-738-PM.dbf TEMP ONLINE

在备库pdbprod8下查看：

SQL> select \* from dba\_temp\_files;

no rows selected

# 坑

## 数据库无法与磁盘组资源建立联系

Duplicate时rman报错通道异常终止，而alert报错如下：

ERROR: failed to establish dependency between database sgerp5 and diskgroup resource ora.DATA.dg

解决办法，手动添加资源：

$ srvctl modify database -d <db\_name> -a "<diskgroup\_list>"

For example:

$ srvctl modify database -d db112 -a "DATA,FRA"

## ORA-01033和ORA-16191

使用如下sql检查错误时：

select DEST\_NAME,STATUS,ERROR,to\_char(FAIL\_DATE,'yyyymmdd hh24:mi:ss') from v$archive\_dest order by 4 desc;

当出现如下俩个错误时（肯定不是同时出现）：

ORA-01033: ORACLE initialization or shutdown in progress 20181102 13:14:47

ORA-16191: Primary log shipping client not logged on standby

把没报错的节点的密码文件拷一份覆盖报错的节点的密码文件。

该报错就是有节点的密码文件不一致导致的。

所以搭建dg的时候，最好是所有相关节点（主库+备库）使用同一份密码文件（通过copy），否则，即使用orapwd创建密码完全一样的密码文件也不解决问题。

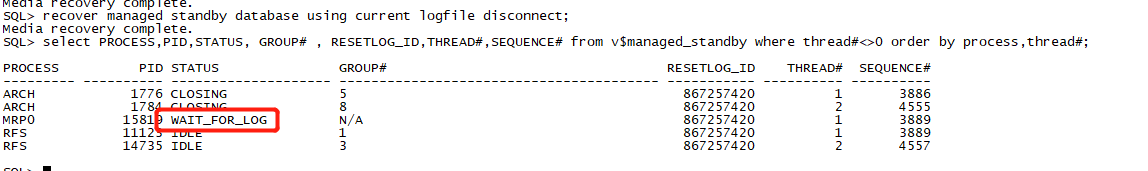
## Standby log太小

源库的redolog是512M的，备库创建standby log创建的是500M，导致ADG可以同步日志但是不能实时同步日志。

如果没有创建standby log，在应用日志的时候就会报错。

但是如果standby log创建小了，在应用日志的时候不会报错，但是在实时应用的时候会有问题。

Mrp0状态如下：



Alert日志中，在[RFS]下会报如下错误：

RFS[1]: No standby redo logfiles available for thread 1

将standby log重建，大于等于源库最大的redo log即可。

## Duplicate时出bug不成功

手动完成duplicate要做的事情就可以了：

1. 备库控制文件恢复好
2. 备库参数改好
3. 然后执行如下脚本

#!/bin/bash

rman target sys/crm2bk\_oracle@crm2bk\_OLD auxiliary sys/crm2bk\_oracle@crm2bk\_NEW log=/home/oracle/enmo/yx01.log<<EOF

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;

set newname for tempfile 1 to

"+gsdxqxdb\_data";

switch clone tempfile all;

set newname for datafile 1 to

"+gsdxqxdb\_data";

set newname for datafile 2 to

"+gsdxqxdb\_data";

set newname for datafile 3 to

"+gsdxqxdb\_data";

……

set newname for datafile 42 to

"+gsdxqxdb\_data";

set newname for datafile 43 to

"+gsdxqxdb\_data";

set newname for datafile 44 to

"+gsdxqxdb\_data";

backup as copy reuse

datafile 1 auxiliary format "+gsdxqxdb\_data"

datafile 2 auxiliary format "+gsdxqxdb\_data"

datafile 3 auxiliary format "+gsdxqxdb\_data"

……

datafile 42 auxiliary format "+gsdxqxdb\_data"

datafile 43 auxiliary format "+gsdxqxdb\_data"

datafile 44 auxiliary format "+gsdxqxdb\_data" ;

}

exit

EOF

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*duplicate is over!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*">>/home/oracle/enmo/rman01.log

所有数据文件，一个不能少。

然后完事儿以后，一定要在备库上：

把新的数据文件catalog到控制文件，然后进行一次switch database to copy。