

Oracle9i+RMAN+DataGuard 实施手记

郭岳(CNOUG ID:David.Guo ITPUB ID:gytyl)

Blog: <http://www.oracledba.com.cn>

原文来自 www.oracledba.com.cn,如需转载,请勿更改文档任何部分

一:硬件环境

主机环境:IBM P570,1.65G CPU2 颗,内存 4GB,本地硬盘 73GB*2,DS4300 阵列,硬盘 300GB*14

备机环境: IBM P570,1.65G CPU2 颗,内存 4GB,本地硬盘 73GB*2,DS4300 阵列,硬盘 300GB*14

二:软件环境

主机环境:AIX 5.3 ML03,Oracle 9.2.0.7.0.

备机环境:AIX5.3 ML03,Oracle 9.2.0.7.0

三:实施过程

3.1:切换主数据库到归档模式

首先,生成一个 pfile,然后,使用命令 `shutdown immediate`,将数据库关闭,切记在这步操作不可以使用 `shutdown abort`,因为如果使用了 `abort` 模式关闭主库,则在启动的时候,无法将数据库启动到归档模式,会提示数据库需要进行 `recover`.

关闭数据库以后,修改 pfile,将其中的参数 log_archive_start 设置为 true,并且设置好归档路径 log_archive_dest,然后从 pfile 将数据库 mount 起来.

Mount 数据库以后,执行 alter database archivelog;将数据库启动到归档模式

3.2:备份主数据库

数据库到归档模式以后,要作的就是备份主数据库了.因为主数据库的所有数据文件都是裸设备,因此必须使用 rman 来进行备份.

在主库上,用 rman 启动 rman 命令,因为我们的备份不使用 catalog,因此直接使用控制文件来代替 catalog.

在 rman 提示符下,用如下命令备份整个数据库

```
RMAN> run {  
  
2> allocate channel dev1 type disk;  
  
3> set limit channel dev1 kbytes 10000000;  
  
4> backup full database format '/archive_log/backup_%t%s';  
  
5> }
```

数据库备份后, 将所有的备份文件拷贝到备机. 注意如果是 ftp 模式, 必须使用 bin 的模式, 否则可能会导致文件损坏而无法恢复.

3.3:备库创建裸设备

在备库上要创建和主库一样的裸设备, 有个简单的方法.

在备库上, 执行:

```
lsvg -l datavg >/tmp/mklv.sh
```

其中的记录应该是类似这样的:

```
system jfs 8 8 1 open/syncd N/A
```

然后, 将这个脚本修改为类似:

```
mklv -y system -w n -s n -r n -t jfs datavg 8
```

的形式, 然后执行之,

裸设备创建好以后, 要修改裸设备的属主和权限

修改属主使用类似如下的语句执行:

```
chown oracle:dba /dev/rssystem
```

修改权限使用类似如下的语句执行:

```
chmod 777 /dev/rssystem
```

执行完成这些以后, 需要仔细检查和主库的是否一样, 否则会导致在 rman 恢复过程中无法完成.

3.4: 创建 standby controlfile 以及 pfile

在主库上, 登陆到 sqlplus 中

执行以下语句:

```
SQL>connect /as sysdba;
```

```
SQL>alter database create standby controlfile to  
'/backup/standby.ctl';
```

```
SQL>create pfile='/backup/pfile.ora' from spfile;
```

然后将生成的控制文件和 pfile 文件传输到备库,并修改好 pfile 文件.

以下贴出备库的 pfile 的内容:

```
*.background_dump_dest='/oracle/product/9.2.0/admin/oradb/bdump'  
*.compatible='9.2.0.0.0'  
*.control_files='/oracle/standby1.ctl','/oracle/standby2.ctl','/oracle/standby3.ctl'  
*.core_dump_dest='/oracle/product/9.2.0/admin/oradb/cdump'  
*.db_block_size=8192  
*.db_cache_size=1572864000  
*.db_domain=""  
*.db_file_multiblock_read_count=16  
*.db_name='oradb'  
*.fast_start_mtrr_target=300  
*.global_names=TRUE  
*.hash_join_enabled=TRUE  
*.instance_name='oradb'  
*.java_pool_size=52428800  
*.job_queue_processes=20  
*.large_pool_size=52428800  
#*.log_archive_dest='/archive_log'  
*.log_archive_start=true  
*.open_cursors=300  
*.open_links=15  
*.open_links_per_instance=15  
*.pga_aggregate_target=209715200  
*.processes=150  
*.query_rewrite_enabled='FALSE'  
*.remote_login_passwordfile='EXCLUSIVE'  
*.service_names='ORADB'  
*.shared_pool_size=209715200  
*.sort_area_size=1048576  
*.star_transformation_enabled='FALSE'  
*.timed_statistics=TRUE  
*.undo_management='AUTO'  
*.undo_retention=10800  
*.undo_tablespace='UNDOTBS1'  
*.user_dump_dest='/oracle/product/9.2.0/admin/oradb/udump'  
*.log_archive_dest_1='LOCATION=/archive_log'  
*.log_archive_dest_state_1=enable  
*.log_archive_format=%t_%s.dbf  
*.log_archive_start=true
```

```
*.standby_archive_dest='/archive_log'  
*.standby_file_management=auto  
*.fal_server=standby  
*.fal_client=primary  
*.remote_archive_enable=true
```

同时,将主库的 pfile 也修改一下,要配置一定的 data guard 的参数,

以下为主库的 pfile 的内容

```
*.background_dump_dest='/oracle/product/9.2.0/admin/oradb/bdump'  
*.compatible='9.2.0.0.0'  
*.control_files='/dev/rconctl01','/dev/rconctl02','/dev/rconctl03'  
*.core_dump_dest='/oracle/product/9.2.0/admin/oradb/cdump'  
*.db_block_size=8192  
*.db_cache_size=1572864000  
*.db_domain=""  
*.db_file_multiblock_read_count=16  
*.db_name='oradb'  
*.fast_start_mtr_target=300  
*.global_names=TRUE  
*.hash_join_enabled=TRUE  
*.instance_name='oradb'  
*.java_pool_size=52428800  
*.job_queue_processes=20  
*.large_pool_size=52428800  
#*.log_archive_dest='/archive_log'  
*.log_archive_start=true  
*.open_cursors=300  
*.open_links=15  
*.open_links_per_instance=15  
*.pga_aggregate_target=209715200  
*.processes=150  
*.query_rewrite_enabled='FALSE'  
*.remote_login_passwordfile='EXCLUSIVE'  
*.service_names='ORADB'  
*.shared_pool_size=209715200  
*.sort_area_size=1048576  
*.star_transformation_enabled='FALSE'  
*.timed_statistics=TRUE  
*.undo_management='AUTO'  
*.undo_retention=10800  
*.undo_tablespace='UNDOTBS1'  
*.user_dump_dest='/oracle/product/9.2.0/admin/oradb/udump'  
*.log_archive_dest_1='LOCATION=/archive_log'
```

```
*.log_archive_dest_2='SERVICE=standby reopen=60'  
*.log_archive_dest_state_1=enable  
*.log_archive_dest_state_2=enable  
*.log_archive_format=%t_%s.dbf  
*.log_archive_start=true  
*.remote_archive_enable=true
```

都修改好以后,就可以在备库上作 restore 了.

3.5:备库作恢复

在备库上,将数据库启动到 nomount 状态

然后用

```
SQL>alter database mount standby database
```

启动备库到 standby 模式.

试用 rman 作恢复,注意,因为备库放备份的地方要和主库一致.才可以正常恢复,否则需要修改参数,才可以作恢复.

恢复的命令如下:

```
RMAN>restore database
```

执行时间会比较久,在全部执行完成后.稍微等待几分钟,在主库执行如下的 sql:

```
SQL>select status, error from v$archive_dest where dest_id=2;
```

结果必须为 valid,表示主库到备库的 log 归档正确,可以在稍等后,去查看备库的归档路径上是否在 rman 过程中,以及备份文件传输以及 restore 过程中产生的 log 是否自动过来.如果过来了,在备库执行一下语句作 recover

```
SQL>recover automatic standby database;
```

等所有的日志都 recover 以后,将备库启动到 recover 模式

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

此时备库将自动应用在 recover 模式,可以通过查询 v\$archived_log 来看新过来的 log 有否自动应用.

一般可试用如下语句查看:

```
SQL>SELECT * FROM v$archived_log WHERE applied='YES';
```

查询的结果,即为已经在备库成功应用的 log.

3.5:清理已经应用过的 log

在主库和备库上,都会产生大量的归档 log,这些 log 在应用到备库后,就需要删除,我们的删除方法试用了 crontab 的自动执行的模式.

在 备 库 , 有 脚 本
get_archive_log.sql,get_archive_log.sh,rm_archive_log.sh

这三个脚本的内容如下:

Get_archvie_log.sql 内容如下:目的是为了得到已经应用的 log 的名称

```
sqlplus -S /nolog <<EOF
connect / as sysdba
set heading off
set echo off
set feedback off
spool /archive_log/rmlog.sh
SELECT 'rm -f '''1_||to_char(SEQUENCE#)||'.dbf' FROM v$archived_log WHERE
applied='YES';
spool off
```

get_archive_log.sh 内容如下,目的是为了执行上面的 sql

```
su - oracle -c "/archive_log/get_archive_log.sql"
```

rm_archive_log.sh 内容如下,目的是作删除

```
cd /archive_log  
./get_archive_log.sh  
./rmlog.sh
```

主机上有两个脚本 getrmlog.sh 以及 rm_archive_log.sh

getrmlog.sh 的内容如下,目的是为了从备机上得到 rmlog.sh 文件,因为日志是否应用,备机上是最准确的,因此从备机去获得 rmlog.sh 文件过来执行.

```
ftp -i -in <<!  
open 10.121.8.99  
user root rootpass  
lcd /archive_log  
cd /archive_log  
get rmlog.sh  
bye
```

rm_archive_log.sh 内容如下,目的是为了执行删除工作.

```
cd /archive_log  
./getrmlog.sh  
./rmlog.sh
```

然后在主库的 crontab 中增加如下的内容:

```
0 7 * * * /archive_log/rm_archive_log.sh
```

表示每天早上 7 点清除已经应用的 log

在备库的 crontab 中增加如下的内容:

```
0 6 * * * /archive_log/rm_archive_log.sh
```

表示每天早上 6 点清除已经应用的 log,备库必须比主库先开始,因为主库要去获得备库的 rmlog.sh 来执行.

3.6:日常检查工作

对于 data guard 数据库,由于备库一直出于 recover 模式,因此,从外部是无法连接的.

日常检查可以从以下几个方面进行

1:可以查看主库和备库的/archive_log 目录中的 log 是否正常删除了,如果没有,则有可能是因为日志没有应用导致的,如果主库上有大量 log,备库没有,则可能是网络问题导致主库的 log 没有正常应用到备库.

2:在备库上依次执行如下操作:

```
SQL>connect /as sysdba
```

```
SQL>recover managed standby database cancel;
```

```
SQL>alter database open read only;
```

然后进行查询,看主库最近的应用数据有没有在备库中.

执行了上面的操作,可以通过下面的操作将备库继续运行到 recover 模式

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

以上为简单的记录,特别感谢巴乔和 piner 的支持,欢迎讨论

Email:guoyue@gmail.com