

【DATAGUARD】 将 11g 物理备库转换为 Snapshot Standby

BLOG 文档结构图

| |
|---|
| 1.1.1.1 将 11g 物理备库转换为 Snapshot Standby |
| 一、将物理备库转换为 Snapshot Standby |
| 1、停止 Redo Apply，让物理备库处于 mounted 状态 |
| 2、确保闪回恢复区已指定 |
| 3、调整备库到 Snapshot Standby 数据库状态 |
| 4、将备库置于对外可读写状态 |
| 5、测试备库处于 Snapshot Standby 数据库对主库日志的接收 |
| 6、在 Snapshot Standby 数据创建用户和表并初始化数据 |
| 二、恢复 Snapshot Standby 数据库为 Physical Standby 数据库 |
| 1、重启备库到 MOUNTED 状态 |
| 2、一条命令恢复原物理备库身份 |
| 3、备库的 alert 日志清楚的记录了这个切换的过程 |
| 4、重启备库到自动恢复日志状态 |
| 5、开启备库到 READ ONLY 状态验证之前在 Snapshot Standb... |
| 6、检查主备库日志是否同步 |
| 三、小结 |

1.1.1 将 11g 物理备库转换为 Snapshot Standby

Oracle 11g 的 Data Guard 不仅仅带给我们的是 Active Data Guard 实时查询特性，同时还带来了另外一个惊喜，这便是 Snapshot Standby 数据库功能，此项功能可将备库置身于“可读写状态”用于不方便在生产环境主库中测试的内容，比如模拟上线测试等任务。当备库读写状态下任务完成后，可以非常轻松的完成 Snapshot Standby 数据库角色切换回备库角色，恢复与主库数据同步。在 Snapshot Standby 数据库状态下，备库是可以接受主库传过来的日志，但是不能够将变化应用在备库中。当 physical standby 数据库转换为 snapshot standby 数据库时，它是一个完全可更新的 standby 数据库。snapshot standby 数据库接收和归档来自 primary 数据库的 redo 数据，但是它不会应用。当 snapshot standby 数据库转换为 physical standby 数据库时，所有在 snapshot standby 数据库的操作被丢弃之后，physical standby 数据库才会应用 primary 数据库的 redo 数据。

一般情况下，物理 standby 数据库处于 mount 状态接收和应用主库的 REDO 日志，物理 standby 数据库不能对外提供访问。如果需要只读访问，那么可以临时以 read-only 的方式 open 物理备库，或者配置 ACTIVE DATA GUARD，那么物理 standby 数据库可以进行只读 (read-only) 访问（比如报表业务查询），但是物理 standby 数据库不能进行读写操作 (read-write)。

有些情况下，为了实现系统的压力测试或者 Real Application Testing (RAT) 或者其他读写操作测试，那么可以临时将物理 standby 数据库转换为 snapshot standby 数据库然后进行测试，因为 snapshot standby 数据库是独立于主库的，并且是可以进行读写操作 (read-write)。测试过程中 snapshot standby 数据库正常接收主库的归档日志，保证主库的数据安全，但是不会应用这些日志，当压力测试结束后，可以非常简单的再将 snapshot standby 转换为物理 standby 数据库，继续同步主库日志

一、 将物理备库转换为 Snapshot Standby

1、 停止 Redo Apply，让物理备库处于 mounted 状态

如果备库正处于 Redo Apply 过程，需要先取消。

```
[oracle@rhel6_lhr ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Tue Apr 7 14:35:29 2015

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

14:35:29 SQL> show parameter name

NAME                                TYPE                                VALUE
-----                                -                                -
db_file_name_convert                string                             oradg11g, oradgss
db_name                             string                             oradg11g
db_unique_name                      string                             oradgss
global_names                       boolean                           FALSE
instance_name                      string                             oradgss
lock_name_space                    string
log_file_name_convert              string                             oradg11g, oradgss
processor_group_name               string
service_names                     string                             oradgss
14:35:35 SQL> select dbid,name,current_scn,protection_mode,protection_level,database_role,force_logging,open_mode,switchover_status from v$database;

   DBID NAME          CURRENT_SCN PROTECTION_MODE      PROTECTION_LEVEL    DATABASE_ROLE    FOR OPEN_MODE          SWITCHOVER_STATUS
-----
1403587593 ORADG11G      1192725 MAXIMUM PERFORMANCE  MAXIMUM PERFORMANCE  PHYSICAL STANDBY YES  READ ONLY WITH APPLY  NOT ALLOWED

Elapsed: 00:00:00.00
14:36:03 SQL> ! ps -ef|grep ora_mrp0_oradgss
oracle      5073      1  0 14:26 ?          00:00:00 ora_mrp0_oradgss
oracle      5726    5655  0 14:36 pts/12    00:00:00 /bin/bash -c ps -ef|grep ora_mrp0_oradgss
oracle      5728    5726  0 14:36 pts/12    00:00:00 grep ora_mrp0_oradgss

14:36:59 SQL> alter database recover managed standby database cancel;

Database altered.

Elapsed: 00:00:01.00
14:37:20 SQL> alter database close;

Database altered.

Elapsed: 00:00:00.06
14:37:35 SQL> select database_role,open_mode from v$database;

DATABASE_ROLE    OPEN_MODE
-----
PHYSICAL STANDBY MOUNTED

Elapsed: 00:00:00.00
14:37:59 SQL>
```

2、 确保闪回恢复区已指定

实现 Snapshot Standby 数据库功能并不需要开启主库和备库的闪回数据库（Flashback Database）功能，与是否开启闪回数据库无关。

```
14:37:35 SQL> select database_role,open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
PHYSICAL STANDBY MOUNTED

Elapsed: 00:00:00.00
14:37:59 SQL> show parameter db_recovery_file_dest

NAME                                TYPE                VALUE
-----
db_recovery_file_dest               string              /u01/app/oracle/flash_recovery
area
db_recovery_file_dest_size          big integer         4122M
14:40:03 SQL> select FLASHBACK_ON from v$database;

FLASHBACK_ON
-----
NO

Elapsed: 00:00:00.01
14:40:32 SQL>
```

3、 调整备库到 Snapshot Standby 数据库状态

只需要执行一条非常简单的 SQL 命令便可以将备库调整到 Snapshot Standby 数据库。

```
14:40:32 SQL> alter database convert to snapshot standby;

Database altered.

Elapsed: 00:00:03.40
14:42:16 SQL> select database_role,open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
SNAPSHOT STANDBY MOUNTED

Elapsed: 00:00:00.01
14:42:50 SQL>
```

分析切换过程中的日志信息

ora11g 主库 alert 日志：

```
Tue Apr 07 14:42:12 2015
alter database convert to snapshot standby
Starting background process RVWR
Tue Apr 07 14:42:12 2015
RVWR started with pid=24, OS id=6015
Allocated 3981120 bytes in shared pool for flashback generation buffer
Created guaranteed restore point SNAPSHOT_STANDBY_REQUIRED_04/07/2015 14:42:12
krsv_proc_kill: Killing 4 processes (all RFS)
Begin: Standby Redo Logfile archival
End: Standby Redo Logfile archival
```

```
RESETLOGS after incomplete recovery UNTIL CHANGE 1192796
Resetting resetlogs activation ID 1403546633 (0x53a86c09)
Online log /u01/app/oracle/oradata/oradgss/redo01.log: Thread 1 Group 1 was previously cleared
Online log /u01/app/oracle/oradata/oradgss/redo02.log: Thread 1 Group 2 was previously cleared
Online log /u01/app/oracle/oradata/oradgss/redo03.log: Thread 1 Group 3 was previously cleared
Standby became primary SCN: 1192794
Tue Apr 07 14:42:16 2015
Setting recovery target incarnation to 3
AUDIT_TRAIL initialization parameter is changed back to its original value as specified in the parameter file.
CONVERT TO SNAPSHOT STANDBY: Complete - Database mounted as snapshot standby
Completed: alter database convert to snapshot standby
关键的一行提示信息“Created guaranteed restore point SNAPSHOT_STANDBY_REQUIRED_03/19/2012 18:46:26”，这里给出了我们转换成 snapshot 的时刻，便于后面的回切。
```

4、 将备库置于对外可读写状态

```
14:42:50 SQL> alter database open;

Database altered.

Elapsed: 00:00:06.09
14:43:55 SQL> select database_role,open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
SNAPSHOT STANDBY  READ WRITE

Elapsed: 00:00:00.01
14:44:45 SQL>
```

5、 测试备库处于 Snapshot Standby 数据库对主库日志的接收

当主库切换日志时，备库依然可以接收到日志，只是并不应用

1) 主库切换日志

```
14:32:58 SQL> alter system switch logfile;

System altered.

Elapsed: 00:00:00.01
14:50:04 SQL>
```

2) 主库记录的 alert 日志内容

ora11g 主库 alert 日志:

```
Tue Apr 07 14:50:04 2015
LNS: Standby redo logfile selected for thread 1 sequence 78 for destination LOG_ARCHIVE_DEST_4
```

ora11gdg 备库 alert 日志:

```
Tue Apr 07 14:50:04 2015
RFS[5]: Selected log 4 for thread 1 sequence 78 dbid 1403587593 branch 876067148
Tue Apr 07 14:50:04 2015
Archived Log entry 8 added for thread 1 sequence 77 ID 0x53a86c09 dest 1:
```

3) 查看主库和备库归档目录下的日志文件内容

(1) 主库归档日志文件

```
[oracle@rhel6_lhr ~]$ ls -ltr /u01/app/oracle/flash_recovery_area/ORADG11G/archivelog/2015_04_07/
total 41236
。。。。。省略。。。。。
-rw-r----- 1 oracle asmadmin    38400 Apr  7 14:20 o1_mf_1_74_bl6xszyqy_.arc
-rw-r----- 1 oracle asmadmin   242176 Apr  7 14:27 o1_mf_1_75_bl6y75kr_.arc
-rw-r----- 1 oracle asmadmin   889344 Apr  7 14:47 o1_mf_1_76_bl6zdp5q_.arc
-rw-r----- 1 oracle asmadmin    69120 Apr  7 14:50 o1_mf_1_77_bl6zjwor_.arc
```

(2) 备库归档日志文件

```
[oracle@rhel6_lhr ~]$ ls -ltr /u01/app/oracle/flash_recovery_area/ORADGSS/archivelog/2015_04_07/
total 14004
-rw-r----- 1 oracle asmadmin 12754944 Apr  7 14:05 o1_mf_1_71_0mq3pp4i_.arc
-rw-r----- 1 oracle asmadmin   251904 Apr  7 14:27 o1_mf_1_72_bl6y72pq_.arc
-rw-r----- 1 oracle asmadmin    38912 Apr  7 14:27 o1_mf_1_73_bl6y72nz_.arc
-rw-r----- 1 oracle asmadmin    38400 Apr  7 14:27 o1_mf_1_74_bl6y72wy_.arc
-rw-r----- 1 oracle asmadmin   242176 Apr  7 14:27 o1_mf_1_75_bl6y765n_.arc
-rw-r----- 1 oracle asmadmin    38400 Apr  7 14:43 o1_mf_1_1_bl6z598f_.arc
-rw-r----- 1 oracle asmadmin   889344 Apr  7 14:47 o1_mf_1_76_bl6zdpk0_.arc
-rw-r----- 1 oracle asmadmin    69120 Apr  7 14:50 o1_mf_1_77_bl6zjwql_.arc
```

可见，备库已经接受到主库发过来的日志。

6、在 Snapshot Standby 数据创建用户和表并初始化数据

```
14:44:45 SQL> create user TEST identified by test;
User created.

Elapsed: 00:00:00.81
14:55:12 SQL> grant dba to test;
Grant succeeded.

Elapsed: 00:00:00.05
14:55:17 SQL> create table test.test as select * from user_tables;
Table created.

Elapsed: 00:00:02.31
14:55:42 SQL> select count(1) from test.test;

COUNT(1)
```

```
-----
          984
Elapsed: 00:00:00.00
14:55:59 SQL>

--源库有 1hr 用户这里我们删除

14:59:18 SQL> drop user 1hr cascade;

User dropped.

Elapsed: 00:00:11.51
14:59:39 SQL>
```

结论，此时备库是一个可任意修改和调整的状态，也就是我们要的“READ WRITE”可读写状态。
特别注意的是，原理上实现 Snapshot Standby 数据库功能是基于闪回数据原理的，因此任何导致闪回数据库无法回退的动作在这里也要规避，否则 Snapshot Standby 数据库将无法回到曾经的备库恢复状态。

二、 恢复 Snapshot Standby 数据库为 Physical Standby 数据库

1、 重启备库到 MOUNTED 状态

```
14:59:39 SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
15:00:41 SQL> startup mount
ORACLE instance started.

Total System Global Area  417546240 bytes
Fixed Size                  2228944 bytes
Variable Size              243273008 bytes
Database Buffers           163577856 bytes
Redo Buffers                8466432 bytes
Database mounted.
15:00:57 SQL> select database_role,open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
SNAPSHOT STANDBY MOUNTED

Elapsed: 00:00:00.01
15:01:00 SQL>
```

2、 一条命令恢复原物理备库身份

```
15:01:00 SQL> alter database convert to physical standby;

Database altered.

Elapsed: 00:00:09.42
15:01:58 SQL>
```

3、备库的 alert 日志清楚的记录了这个切换的过程

```
Tue Apr 07 15:01:48 2015
alter database convert to physical standby
ALTER DATABASE CONVERT TO PHYSICAL STANDBY (oradgss)
Flashback Restore Start
Flashback Restore Complete
Drop guaranteed restore point
Stopping background process RVWR
Deleted Oracle managed file /u01/app/oracle/flash_recovery_area/ORADGSS/flashback/o1_mf_bl6z24yh_.flb
Deleted Oracle managed file /u01/app/oracle/flash_recovery_area/ORADGSS/flashback/o1_mf_bl6z28jq_.flb
Guaranteed restore point dropped
Clearing standby activation ID 1403924465 (0x53ae2ff1)
The primary database controlfile was created using the
'MAXLOGFILES 16' clause.
There is space for up to 13 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 'srl1.f' SIZE 52428800;
ALTER DATABASE ADD STANDBY LOGFILE 'srl2.f' SIZE 52428800;
ALTER DATABASE ADD STANDBY LOGFILE 'srl3.f' SIZE 52428800;
ALTER DATABASE ADD STANDBY LOGFILE 'srl4.f' SIZE 52428800;
Shutting down archive processes
Archiving is disabled
Tue Apr 07 15:01:50 2015
ARCH shutting down
ARC3: Archival stopped
Tue Apr 07 15:01:50 2015
ARCH shutting down
ARC1: Archival stopped
Tue Apr 07 15:01:50 2015
ARCH shutting down
ARC0: Archival stopped
Completed: alter database convert to physical standby
```

从 alert 日志中可以得到恢复方法使用的闪回数据库功能实现的，也就是说，即便备库没有运行在闪回数据库状态，依然可以使用闪回数据库功能完成备库的角色转换。

4、重启备库到自动恢复日志状态

```
(1) 此时数据库处于 NOMOUNTED 状态，需要重新启动数据库。
注意这里是重启数据库，而不是使用 alter 命令调整，否则会收到如下报错：
SQL > alter database mount;
alter database mount
*
ERROR at line 1:
```


ORA-00750: database has been previously mounted and dismounted

```
15:01:58 SQL> select database_role,open_mode from v$database;
select database_role,open_mode from v$database
*
```

ERROR at line 1:
ORA-01507: database not mounted

Elapsed: 00:00:00.01
15:04:57 SQL> shutdown immediate;
ORA-01507: database not mounted

ORACLE instance shut down.
15:05:14 SQL> startup mount;
ORACLE instance started.

```
Total System Global Area  417546240 bytes
Fixed Size                  2228944 bytes
Variable Size               243273008 bytes
Database Buffers           163577856 bytes
Redo Buffers                 8466432 bytes
```

Database mounted.
15:05:25 SQL> alter database recover managed standby database using current logfile disconnect from session;

Database altered.

Elapsed: 00:00:06.02
15:06:26 SQL>

(2) 查看备库 alert 日志，可以清楚的看到恢复的过程。

```
alter database recover managed standby database using current logfile disconnect from session
```

```
Attempt to start background Managed Standby Recovery process (oradgss)
```

```
Tue Apr 07 15:06:20 2015
```

```
MRP0 started with pid=27, OS id=7673
```

```
MRP0: Background Managed Standby Recovery process started (oradgss)
```

```
started logmerger process
```

```
Tue Apr 07 15:06:25 2015
```

```
Managed Standby Recovery starting Real Time Apply
```

```
Parallel Media Recovery started with 2 slaves
```

```
Waiting for all non-current ORLs to be archived...
```

```
All non-current ORLs have been archived.
```

```
Clearing online redo logfile 1 /u01/app/oracle/oradata/oradgss/redo01.log
```

```
Clearing online log 1 of thread 1 sequence number 79
```

```
Completed: alter database recover managed standby database using current logfile disconnect from session
```

```
Clearing online redo logfile 1 complete
```

```
Clearing online redo logfile 2 /u01/app/oracle/oradata/oradgss/redo02.log
```

```
Clearing online log 2 of thread 1 sequence number 2
```

```
Clearing online redo logfile 2 complete
```

```
Media Recovery Log /u01/app/oracle/flash_recovery_area/ORADGSS/archivelog/2015_04_07/o1_mf_1_76_b16zdpk0_.arc
```

```
Media Recovery Log /u01/app/oracle/flash_recovery_area/ORADGSS/archivelog/2015_04_07/o1_mf_1_77_b16zjwql_.arc
```



```
Media Recovery Log /u01/app/oracle/flash_recovery_area/ORADGSS/archivelog/2015_04_07/o1_mf_1_78_b170ghh8_.arc
Media Recovery Waiting for thread 1 sequence 79 (in transit)
Recovery of Online Redo Log: Thread 1 Group 5 Seq 79 Reading mem 0
  Mem# 0: /u01/app/oracle/oradata/oradgss/standby_redo05.log
```

（3）查看 V\$ARCHIVED_LOG 动态性能视图查看日志应用情况

```
15:06:26 SQL> select sequence#, first_time, next_time, applied from v$archived_log order by sequence#;
```

| SEQUENCE# | FIRST_TIM | NEXT_TIME | APPLIED |
|-----------|-----------|-----------|-----------|
| 1 | 07-APR-15 | 07-APR-15 | NO |
| 71 | 07-APR-15 | 07-APR-15 | YES |
| 72 | 07-APR-15 | 07-APR-15 | YES |
| 73 | 07-APR-15 | 07-APR-15 | YES |
| 74 | 07-APR-15 | 07-APR-15 | YES |
| 75 | 07-APR-15 | 07-APR-15 | YES |
| 76 | 07-APR-15 | 07-APR-15 | YES |
| 77 | 07-APR-15 | 07-APR-15 | YES |
| 78 | 07-APR-15 | 07-APR-15 | IN-MEMORY |

9 rows selected.

```
Elapsed: 00:00:00.01
15:08:56 SQL>
```

5、 **开启备库到 READ ONLY 状态验证之前在 Snapshot Standby 数据库上的操作已撤销**

```
15:08:56 SQL> alter database recover managed standby database cancel;
```

Database altered.

```
Elapsed: 00:00:01.00
```

```
15:11:51 SQL> alter database open read only;
```

Database altered.

```
Elapsed: 00:00:00.21
```

```
15:11:56 SQL> select database_role, open_mode from v$database;
```

| DATABASE_ROLE | OPEN_MODE |
|------------------|-----------|
| PHYSICAL STANDBY | READ ONLY |

```
Elapsed: 00:00:00.01
```

```
15:12:02 SQL> select username from dba_users where username IN ('LHR','TEST');
```

| USERNAME |
|----------|
| LHR |

```
Elapsed: 00:00:00.01
```

```
15:12:42 SQL>
```

之前创建的测试用户 TEST 不存在, 而且被删除的用户 lhr 又被还原了。结论得证。

6、检查主备库日志是否同步

```
主库执行:
15:44:24 SQL> col DEST_NAME format a20
15:44:37 SQL> select ads.dest_id,
15:44:37      2      ads.DEST_NAME,
15:44:38      3      ads.STATUS,
15:44:38      4      ads.TYPE,
15:44:38      5      ads.RECOVERY_MODE,
15:44:38      6      ads.DB_UNIQUE_NAME,
15:44:38      7      max(sequence#) "Current Sequence",
15:44:38      8      max(log_sequence) "Last Archived",
15:44:38      9      from v$archived_log al, v$archive_dest ad, v$archive_dest_status ads
15:44:38     10     where ad.dest_id = al.dest_id
15:44:38     11           and al.dest_id = ads.dest_id
15:44:38     12           and al.resetlogs_change# =
15:44:38     13              (select max(resetlogs_change#) from v$archived_log)
15:44:38     14     group by ads.dest_id,
15:44:38     15              ads.DEST_NAME,
15:44:38     16              ads.STATUS,
15:44:38     17              ads.TYPE,
15:44:38     18              ads.RECOVERY_MODE,
15:44:38     19              ads.DB_UNIQUE_NAME
15:44:38     20     order by ads.dest_id;
```

| DEST_ID | DEST_NAME | STATUS | TYPE | RECOVERY_MODE | DB_UNIQUE_NAME | Current Sequence | Last Archived |
|---------|--------------------|--------|----------|-------------------------|----------------|------------------|---------------|
| 1 | LOG_ARCHIVE_DEST_1 | VALID | LOCAL | IDLE | oradg1lg | 78 | 78 |
| 2 | LOG_ARCHIVE_DEST_2 | VALID | PHYSICAL | MANAGED REAL TIME APPLY | oradgphy | 78 | 79 |
| 3 | LOG_ARCHIVE_DEST_3 | VALID | LOGICAL | LOGICAL REAL TIME APPLY | oradglg | 78 | 79 |
| 4 | LOG_ARCHIVE_DEST_4 | VALID | PHYSICAL | IDLE | oradgss | 78 | 79 |

```
Elapsed: 00:00:00.00
15:44:38 SQL>
```

三、小结

这便是神奇的“Snapshot Standby 数据库”功能，备库可以临时成为一个可读写的独立数据库，这极大的扩展了备库的应用场合，我们可以使用备库的这一项特殊功能将那些在生产环境中“不敢”模拟和再现的问题在备库端进行测试，测试完毕后再恢复其物理备库的身份进行日志恢复。SNAPSHOT STANDBY 模式将备库置于可读写状态，可以在此备库上来回折腾，这个结合 REAL APPLICATION TESTING 做升级前测试非常方便。要注意如果在 SNAPSHOT STANDBY 上面的数据更改操作过大，恢复回 PHYSICAL STANDBY 的时间会非常长。

参考资料：<http://blog.itpub.net/519536/viewspace-719056/>

.....

本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用

ITPUB BLOG : <http://blog.itpub.net/26736162>

本文地址：

QQ：642808185 注明：ITPUB 的文章标题

<版权所有，文章允许转载，但须以链接方式注明源地址，否则追究法律责任!>

.....