**Resolving Gaps in Data Guard Apply Using Incremental RMAN BAckup**

Recently, we had a glitch on a Data Guard (physical standby database) on infrastructure. This is not a critical database; so the monitoring was relatively lax. And that being done by an outsourcer does not help it either. In any case, the laxness resulted in a failure remaining undetected for quite some time and it was eventually discovered only when the customer complained. This standby database is usually opened for read only access from time to time.This time, however, the customer saw that the data was significantly out of sync with primary and raised a red flag. Unfortunately, at this time it had become a rather political issue.  
  
Since the DBA in charge couldn’t resolve the problem, I was called in. In this post, I will describe the issue and how it was resolved. In summary, there are two parts of the problem:  
  
(1) What happened  
(2) How to fix it  
  
**What Happened**  
  
Let’s look at the first question – what caused the standby to lag behind. First, I looked for the current SCN numbers of the primary and standby databases. On the primary:

SQL> select min(fhscn),fhrba\_Seq SEQUENCE from x$kcvfh group by fhrba\_Seq;

MIN(FHSCN) SEQUENCE

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

6525708985273 10185

select distinct((checkpoint\_change#)) from v$datafile\_header order by 1;

select distinct(to\_char(checkpoint\_change#)) from v$datafile\_header order by 1;

SQL> select current\_scn from v$database;  
  
CURRENT\_SCN  
-----------  
1447102

select to\_char(current\_scn) from v$database;

On the standby:  
  
SQL> select current\_scn from v$database;  
  
CURRENT\_SCN  
-----------  
1301571

select to\_char(current\_scn) from v$database;

And check which scn is less use that.

Clearly there is a difference. But this by itself does not indicate a problem; since the standby is expected to lag behind the primary (this is an asynchronous non-real time apply setup). The real question is how much it is lagging in the terms of [wall clock](http://skimlinks.pgpartner.com/mrdr.php?url=http%3A%2F%2Fskimlinks.pgpartner.com%2Fsearch.php%2Fform_keyword%3Dwall%2Bclock). To know that I used the scn\_to\_timestamp function to translate the SCN to a timestamp:  
  
SQL> select scn\_to\_timestamp(1447102) from dual;  
  
SCN\_TO\_TIMESTAMP(1447102)  
-------------------------------  
18-DEC-09 08.54.28.000000000 AM  
  
I ran the same query to know the timestamp associated with the SCN of the standby database as well (note, I ran it on the primary database, though; since it will fail in the standby in a mounted mode):  
  
SQL> select scn\_to\_timestamp(1301571) from dual;  
  
SCN\_TO\_TIMESTAMP(1301571)  
-------------------------------  
15-DEC-09 07.19.27.000000000 PM  
  
This shows that the standby is *two and half days* lagging! The data at this point is not just stale; it must be rotten.  
  
The next question is why it would be lagging so far back in the past. This is a 10.2 database where FAL server should automatically resolved any gaps in archived logs. Something must have happened that caused the FAL (fetch archived log) process to fail. To get that answer, first, I checked the alert log of the standby instance. I found these lines that showed the issue clearly:  
  
…  
Fri Dec 18 06:12:26 2009  
Waiting for all non-current ORLs to be archived...  
Media Recovery Waiting for thread 1 sequence 700  
Fetching gap sequence in thread 1, gap sequence 700-700  
… …  
Fri Dec 18 06:13:27 2009  
FAL[client]: Failed to request gap sequence  
GAP - thread 1 sequence 700-700  
DBID 846390698 branch 697108460  
FAL[client]: All defined FAL servers have been attempted.  
…  
  
Going back in the alert log, I found these lines:  
  
Tue Dec 15 17:16:15 2009  
Fetching gap sequence in thread 1, gap sequence 700-700  
Error 12514 received logging on to the standby  
FAL[client, MRP0]: Error 12514 connecting to DEL1 for fetching gap sequence  
Tue Dec 15 17:16:15 2009  
Errors in file /opt/oracle/admin/DEL2/bdump/del2\_mrp0\_18308.trc:  
ORA-12514: TNS:listener does not currently know of service requested in connect descriptor  
Tue Dec 15 17:16:45 2009  
Error 12514 received logging on to the standby  
FAL[client, MRP0]: Error 12514 connecting to DEL1 for fetching gap sequence  
Tue Dec 15 17:16:45 2009  
Errors in file /opt/oracle/admin/DEL2/bdump/del2\_mrp0\_18308.trc:  
ORA-12514: TNS:listener does not currently know of service requested in connect descriptor  
  
This clearly showed the issue. On December 15th at 17:16:15, the Managed Recovery Process encountered an error while receiving the log information from the primary. The error was ORA-12514 “TNS:listener does not currently know of service requested in connect descriptor”. This is usually the case when the TNS connect string is incorrectly specified. The primary is called DEL1 and there is a connect string called DEL1 in the standby server.  
  
The connect string works well. Actually, right now there is no issue with the standby getting the archived logs; so there connect string is fine - now. The standby is receiving log information from the primary. There must have been some temporary hiccups causing that specific archived log not to travel to the standby. If that log was somehow skipped (could be an intermittent problem), then it should have been picked by the FAL process later on; but that never happened. Since the sequence# 700 was not applied, none of the logs received later – 701, 702 and so on – were applied either. This has caused the standby to lag behind since that time.  
  
So, the fundamental question was why FAL did not fetch the archived log sequence# 700 from the primary. To get to that, I looked into the alert log of the *primary* instance. The following lines were of interest:  
  
…  
Tue Dec 15 19:19:58 2009  
Thread 1 advanced to log sequence 701 (LGWR switch)  
Current log# 2 seq# 701 mem# 0: /u01/oradata/DEL1/onlinelog/o1\_mf\_2\_5bhbkg92\_.log  
Tue Dec 15 19:20:29 2009Errors in file /opt/oracle/product/10gR2/db1/admin/DEL1/bdump/del1\_arc1\_14469.trc:  
ORA-00308: cannot open archived log '/u01/oraback/1\_700\_697108460.dbf'  
ORA-27037: unable to obtain file status  
Linux Error: 2: No such file or directory  
Additional information: 3  
Tue Dec 15 19:20:29 2009  
FAL[server, ARC1]: FAL archive failed, see trace file.  
Tue Dec 15 19:20:29 2009  
Errors in file /opt/oracle/product/10gR2/db1/admin/DEL1/bdump/del1\_arc1\_14469.trc:  
ORA-16055: FAL request rejected  
ARCH: FAL archive failed.  
Archiver continuing  
Tue Dec 15 19:20:29 2009  
ORACLE Instance DEL1 - Archival Error. Archiver continuing.  
…  
  
These lines showed everything clearly. The issue was:  
  
ORA-00308: cannot open archived log '/u01/oraback/1\_700\_697108460.dbf'  
ORA-27037: unable to obtain file status  
Linux Error: 2: No such file or directory  
  
The archived log simply was not available. The process could not see the file and couldn’t get it across to the standby site.  
  
Upon further investigation I found that the DBA actually removed the archived logs to make some room in the filesystem without realizing that his action has removed the most current one which was yet to be transmitted to the remote site. The mystery surrounding why the FAL did not get that log was finally cleared.  
  
**Solution**  
  
Now that I know the cause, the focus was now on the resolution. If the archived log sequence# 700 was available on the primary, I could have easily copied it over to the standby, registered the log file and let the managed recovery process pick it up. But unfortunately, the file was gone and I couldn’t just recreate the file. Until that logfile was applied, the recovery will not move forward. So, what are my options?  
  
One option is of course to recreate the standby - possible one but not technically feasible considering the time required. The other option is to apply the incremental backup of primary from that SCN number. That’s the key – the backup must be from a specific SCN number. I have described the process since it is not very obvious. The following shows the step by step approach for resolving this problem. I have shown where the actions must be performed – [Standby] or [Primary].  
  
1. [Standby] Stop the managed standby apply process:  
  
SQL> alter database recover managed standby database cancel;  
  
Database altered.  
  
2. [Standby] Shutdown the standby database  
  
3. [Primary] On the primary, take an incremental backup from the SCN number where the standby has been stuck:  
  
RMAN> run {   
2> allocate channel c1 type disk format '/u01/oraback/%U.rmb';   
3> backup incremental from scn 1301571 database;  
4> }  
  
using target database control file instead of recovery catalog  
allocated channel: c1  
channel c1: sid=139 devtype=DISK  
  
Starting backup at 18-DEC-09  
channel c1: starting full datafile backupset  
channel c1: specifying datafile(s) in backupset  
input datafile fno=00001 name=/u01/oradata/DEL1/datafile/o1\_mf\_system\_5bhbh59c\_.dbf  
… …  
piece handle=/u01/oraback/06l16u1q\_1\_1.rmb tag=TAG20091218T083619 comment=NONE  
channel c1: backup set complete, elapsed time: 00:00:06  
Finished backup at 18-DEC-09  
released channel: c1  
  
4. [Primary] On the primary, create a new standby controlfile:  
  
SQL> alter database create standby controlfile as '/u01/oraback/DEL1\_standby.ctl';  
  
Database altered.  
  
5. [Primary] Copy these files to standby host:  
  
oracle@oradba1 /u01/oraback# scp \*.rmb \*.ctl oracle@oradba2:/u01/oraback  
oracle@oradba2's password:  
06l16u1q\_1\_1.rmb 100% 43MB 10.7MB/s 00:04   
DEL1\_standby.ctl 100% 43MB 10.7MB/s 00:04   
  
6. [Standby] Bring up the instance in nomount mode:  
  
SQL> startup nomount  
  
7. [Standby] Check the location of the controlfile:  
  
SQL> show parameter control\_files  
  
NAME TYPE VALUE  
------------------------------------ ----------- ------------------------------  
control\_files string /u01/oradata/standby\_cntfile.ctl  
  
8. [Standby] Replace the controlfile with the one you just created in primary.  
  
9. $ cp /u01/oraback/DEL1\_standby.ctl /u01/oradata/standby\_cntfile.ctl  
  
10.[Standby] Mount the standby database:  
  
SQL> alter database mount standby database;  
  
11.[Standby] RMAN does not know about these files yet; so you must let it know – by a process called cataloging. Catalog these files:  
  
$ rman target=/  
  
Recovery Manager: Release 10.2.0.4.0 - Production on Fri Dec 18 06:44:25 2009  
  
Copyright (c) 1982, 2007, Oracle. All rights reserved.  
  
connected to target database: DEL1 (DBID=846390698, not open)  
RMAN> catalog start with '/u01/oraback';  
  
using target database control file instead of recovery catalog  
searching for all files that match the pattern /u01/oraback  
  
List of Files Unknown to the Database  
=====================================  
File Name: /u01/oraback/DEL1\_standby.ctl  
File Name: /u01/oraback/06l16u1q\_1\_1.rmb  
  
Do you really want to catalog the above files (enter YES or NO)? yes  
cataloging files...  
cataloging done  
  
List of Cataloged Files  
=======================  
File Name: /u01/oraback/DEL1\_standby.ctl  
File Name: /u01/oraback/06l16u1q\_1\_1.rmb  
  
12.Recover these files:  
  
RMAN> recover database;  
  
Starting recover at 18-DEC-09  
using channel ORA\_DISK\_1  
channel ORA\_DISK\_1: starting incremental datafile backupset restore  
channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set  
destination for restore of datafile 00001: /u01/oradata/DEL2/datafile/o1\_mf\_system\_5lptww3f\_.dbf  
...…  
channel ORA\_DISK\_1: reading from backup piece /u01/oraback/05l16u03\_1\_1.rmb  
channel ORA\_DISK\_1: restored backup piece 1  
piece handle=/u01/oraback/05l16u03\_1\_1.rmb tag=TAG20091218T083619  
channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:07  
  
starting media recovery  
  
archive log thread 1 sequence 8012 is already on disk as file /u01/oradata/1\_8012\_697108460.dbf  
archive log thread 1 sequence 8013 is already on disk as file /u01/oradata/1\_8013\_697108460.dbf  
… …  
  
13. After some time, the recovery fails with the message:  
  
archive log filename=/u01/oradata/1\_8008\_697108460.dbf thread=1 sequence=8009  
RMAN-00571: ===========================================================  
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============  
RMAN-00571: ===========================================================  
RMAN-03002: failure of recover command at 12/18/2009 06:53:02  
RMAN-11003: failure during parse/execution of SQL statement: alter database recover logfile '/u01/oradata/1\_8008\_697108460.dbf'  
ORA-00310: archived log contains sequence 8008; sequence 8009 required  
ORA-00334: archived log: '/u01/oradata/1\_8008\_697108460.dbf'  
  
This happens because we have come to the last of the archived logs. The expected archived log with sequence# 8008 has not been generated yet.  
  
14.At this point exit RMAN and start managed recovery process:  
  
SQL> alter database recover managed standby database disconnect from session;  
  
Database altered.  
  
15.Check the SCN’s in primary and standby:  
  
[Standby] SQL> select current\_scn from v$database;  
  
CURRENT\_SCN  
-----------  
1447474  
[Primary] SQL> select current\_scn from v$database;  
  
CURRENT\_SCN  
-----------  
1447478  
Now they are very close to each other. The standby has now caught up.