

## Resolving archived log file gaps in Streams

Should your Streams archived log downstream capture environment suffer network outages, causing Log Transport Services to fail, archive log gaps may occur. Streams cannot always resolve gaps automatically depending on the length and frequency of the network outage. In such cases, Streams will fail to mine the “new” archived log files shipped from the source database.

In such a situation it is important to detect and resolve the gap by identifying the missing archived log files from the source database and manually register them on the target database.

The following SQL run on the target database will detect any archived log gaps and provide the missing thread and log sequence number(s):

```
select thread, seq+1 first_seq_missing , seq+(next_seq-seq-1)
last_seq_missing, next_seq-seq-1 missing_count
  from   (select THREAD# thread, SEQUENCE# seq,
              lead (SEQUENCE#, 1, SEQUENCE#) over (partition by
thread# order by sequence#) next_seq
          from   dba_registered_archived_log, dba_capture
          where  capture_name = (select capture_name from
v$streams_capture)
          and   consumer_name = capture_name)
 where  next_seq - seq > 1
 order  by 1,2;
```

We now need to establish the archived log filename from the sequence number. This is possible if the archived log filename contains the log sequence number. I.e The Source database initialisation parameter: log\_archive\_format must contain %s variable.

E.g. log\_archive\_format=arch\_%t\_%s\_%r.log

To find the location of the archived log files, logon to the source database as SYSDBA and execute the following sqlplus command:

```
SQL> show parameter log_archive_dest_1
```

If database is not using ASM, then cd to the location and run the following Unix command:

```
ls arch_thread_<#>_seq_<#>*log
```

If database is using ASM, modify the environment variables for the oracle user and invoke ASM command line utility: asmcmd

N.B. You will need to look in the directory for the date the file was created

E.g.

```
ASMCMDB> cd +FLASH/<db_name>/ARCHIVELOG/<date>
ASMCMDB> ls arch_thread_<#>_seq_<#>*log
```

Having found the missing archived log files on the source database storage, we need to transfer them manually to the target database foreign archived log location. If using ASM, the files need to be copied from ASM to the local file system on the source database server, then copied to the target database server local file system and finally copied to ASM.

The following steps show the process:

1. Using asmcmd utility copy files to /tmp on source database server

E.g.

```
ASMCMD> cp thread_2_seq_4342.8970.694274205 /tmp
ASMCMD> cp thread_2_seq_4343.8972.694274383 /tmp
```

2. From /tmp on source database server copy the files to /tmp on the target database server

E.g.

```
scp /tmp/thread_2_seq_4342.8970.694274205
oracle@<target_hostname>:/tmp
```

```
scp /tmp/thread_2_seq_4343.8972.694274383 /tmp
oracle@<target_hostname>:/tmp
```

3. Using asmcmd utility cd to +FLASH/<db\_name>/FOREIGNARCHIVELOG

4. Using asmcmd utility copy files from /tmp on target database server

E.g.

```
ASMCMD> cp /tmp/thread_2_seq_4342.8970.694274205 .
ASMCMD> cp /tmp/thread_2_seq_4343.8972.694274383 .
```

5. Manually register the foreign archived logs as SYSDBA on the target database.

E.g.

```
SQL> ALTER DATABASE REGISTER LOGFILE
'+FLASH/<db_name>/FOREIGNARCHIVELOG/<logfile spec from above>'
FOR '<capture_name>';
```

6. Finally, logon to target database as Streams Admin user and check the status of the capture process by executing the following query:

```
select CAPTURE_NAME, STATE from V$STREAMS_CAPTURE;
```

The capture process should now have a state of "CAPTURING CHANGES".

## Conclusion

In a RAC environment it is important that the target LOCATION specified on the source database specifies a service that uses the target Virtual IP address. If an instance should fail on the target database and the VIP is not used, Log Transport Services will fail and Streams will stop working until the archived logs for that particular thread are shipped including any gap resolution.

Sometimes the Capture process does not advance and appears stuck in one of the following states:

```
select STATE from V$STREAMS_CAPTURE;
```

shows :

1. INITIALIZING / DICTIONARY INITIALIZATION (the state alternates between these states), or
2. WAITING FOR DICTIONARY REDO: FIRST SCN <SCN> , or
3. WAITING FOR REDO: LAST SCN MINED <SCN> , or
4. WAITING FOR DICTIONARY REDO: FILE <filename>

The problem could be caused by:

- a missing logfile, or
- a logfile is not registered, or
- a logfile is corrupted, or
- Capture process is verifying/prechecking logfiles

The process will remain in this state until the log is located, it is registered or the corruption is resolved (in which case it will also be necessary to reregister the log or the Capture process has checked the logfiles on disk).