Procedure to clear stuck downstream real-time mine processes

Problem Description

On a 2 node RAC database system running Oracle Streams real-time downstream capture, users are complaining that transactions on the source 2 node RAC database are not being replicated to the target.

Solution

Let's check the status of the capture process on the target.

TARGET

Firstly, we must logon to the target database instance that is running the downstream capture (DSC) process.

```
$ sqlplus streams_admin/streams_admin

SQL*Plus: Release 11.1.0.7.0 - Production on Wed May 4 13:09:07 2011

Copyright (c) 1982, 2008, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit
Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options


SQL> col state format a20
SQL> select state, capture_time from v$streams_capture;

STATE                CAPTURE_TIME
-------------------- --------------------
CAPTURING CHANGES    2011-MAY-04 13:10:04
```

All looks well, but checking the database instance alert log on the target (TGT1) reveals a long running transaction.

```
$ view /u01/app/oracle/diag/rdbms/tgt/TGT1/trace/alert_TGT1.log

Wed May 04 12:20:06 2011
CP01: long running txn detected, xid: 0x0027.00f.0009c3ad
Wed May 04 12:30:06 2011
CP01: long running txn detected, xid: 0x0027.00f.0009c3ad
Wed May 04 12:40:06 2011
CP01: long running txn detected, xid: 0x0027.00f.0009c3ad
Wed May 04 12:50:06 2011
CP01: long running txn detected, xid: 0x0027.00f.0009c3ad
Wed May 04 13:00:07 2011
CP01: long running txn detected, xid: 0x0027.00f.0009c3ad
```

This indicates that Streams is waiting for a commit or is actually processing a large transaction.

Streams is not designed to cope with batch updates of several thousand rows. For example, if a column in a table of 1000000 rows on the source database is updated with no WHERE clause, Streams will generate 1000000 LCRs and apply them on the target database. The original DML statement is not replayed. This takes a long time to execute and causes Oracle to write the “long running txn detected” message to the alert log every 10 minutes.

Running a query against v$standby_log reveals six stuck processes from April. Three on thread 1 and three on thread 2. In the example below, today is 2011-MAY-04. So here we have stuck processes, probably caused by a user failing to commit or rollback their transaction.

```
SQL> select GROUP#, DBID, THREAD#, SEQUENCE#, BYTES, USED, ARC, STATUS,
FIRST_TIME from v$standby_log;


    GROUP# DBID             THREAD#  SEQUENCE#      BYTES       USED ARC
STATUS      FIRST_TIME
---------- ------------ ---------- ---------- ---------- ---------- --- ---
-------  --------------------
        20 2066975734              1        102 1073741824   19494400 YES
ACTIVE      2011-APR-11 00:30:10
        21 2066975734              1         97 1073741824  549950976 YES
ACTIVE      2011-APR-08 09:30:53
        22 2066975734              1       1030 1073741824   74392576 YES
ACTIVE      2011-APR-13 00:30:13
        23 2066975734              1       1122 1073741824      34304 YES
ACTIVE      2011-MAY-04 13:11:04
        24 UNASSIGNED              1          0 1073741824        512 NO
UNASSIGNED                0
        25 UNASSIGNED              1          0 1073741824        512 NO
UNASSIGNED                0
        26 UNASSIGNED              1          0 1073741824        512 NO
UNASSIGNED                0
        27 2066975734              2         78 1073741824   35490816 YES
ACTIVE      2011-APR-08 09:30:52
        28 2066975734              2         83 1073741824   13549056 YES
ACTIVE      2011-APR-11 00:30:12
        29 2066975734              2        938 1073741824   44955136 YES
ACTIVE      2011-APR-13 00:30:13
        30 2066975734              2       1004 1073741824      46592 YES
ACTIVE      2011-MAY-04 13:11:04
        31 UNASSIGNED              2          0 1073741824        512 NO
UNASSIGNED                0
        32 UNASSIGNED              2          0 1073741824        512 NO
UNASSIGNED                0
        33 UNASSIGNED              2          0 1073741824        512 YES
UNASSIGNED                0

14 rows selected.
```

Stopping and starting Streams DSC does not clear the problem. To clear the "downstream real-time mine" stuck processes, we must transition the Streams capture process to "downstream capture" by executing the following procedure:

```
SQL> BEGIN
```

```
  2
  3        dbms_output.put_line('Switching capture processes to real-
time mine ...');
  4        dbms_capture_adm.set_parameter( capture_name =>
'SRC_SCHEMA_CAPTURE',
  5                                        parameter    =>
'downstream_real_time_mine',
  6                                        VALUE        => 'n');
  7
end;
/

PL/SQL procedure successfully completed.
```

The procedure will automatically stop and start Streams. Note the VALUE parameter has to have a value of 'n'.

SOURCE

Logon to a source database instance and perform a logfile switch:

```
$ sqlplus / as sysdba

SQL> alter system archive log current;

System altered.
```

TARGET

Log back onto the target database instance running DSC and transition Streams back from "downstream capture" to "downstream real-time mine" by executing the following procedure:

```
$ sqlplus streams_admin/password

SQL*Plus: Release 11.1.0.7.0 - Production on Wed May 4 13:14:57 2011

Copyright (c) 1982, 2008, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options

SQL> BEGIN
  2
  3        dbms_output.put_line('Switching capture processes to real-time
mine ...');
  4        dbms_capture_adm.set_parameter( capture_name =>
'SRC_SCHEMA_CAPTURE',
  5                                        parameter    =>
'downstream_real_time_mine',
  6                                        VALUE        => 'y');
  7
end;
/

PL/SQL procedure successfully completed.
```

The procedure will automatically stop and start Streams. Note the VALUE parameter has to have a value of 'y'.

Now query v$standby_log to confirm that the stuck processes have cleared.

```
SQL> select GROUP#, DBID, THREAD#, SEQUENCE#, BYTES, USED, ARC, STATUS,
FIRST_TIME from v$standby_log;

    GROUP# DBID             THREAD#  SEQUENCE#      BYTES       USED ARC
STATUS     FIRST_TIME
---------- ------------ ---------- ---------- ---------- ---------- --- ---
------- --------------------
        20 2066975734             1       1125 1073741824     747520 YES
ACTIVE     2011-MAY-04 13:15:23
        21 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        22 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        23 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        24 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        25 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        26 UNASSIGNED             1          0 1073741824        512 YES
UNASSIGNED               0
        27 2066975734             2       1007 1073741824     784896 YES
ACTIVE     2011-MAY-04 13:15:23
        28 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0
        29 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0
        30 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0
        31 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0
        32 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0
        33 UNASSIGNED             2          0 1073741824        512 YES
UNASSIGNED               0

14 rows selected.
```

Finally check the capture process is “capturing changes”.

```
SQL> select CAPTURE_NAME, STATE from v$streams_capture;

CAPTURE_NAME            STATE
------------------      ------------------
SRC_SCHEMA_CAPTURE      CAPTURING CHANGES
```

Gotcha

It is important to perform a logfile switch on the source database to “kick” Streams into a given mode after reconfiguration. E.g. Transitioning from DSC to real-time mine and vice versa.