Oracle Database Administartion Services

KE\_Oracle\_Atlas replatform tablespace readonly issues

Submitted to

****

**By**



CIS, Wipro Limited

Document Details

|  |  |
| --- | --- |
| Project Name | Innogy SE |
| Account | CIS |
| IT Component/Application Title | Oracle\_Atlas replatform tablespace readonly issues |
| Current Version | 1.2 |
| List of Contributors | Kamlesh Sawant , Chetan Patel , Saurabh Nigam , Abhishek Kansal , Jiten Pansara,SreyaPuthukudy |
| Customer Contact Information |  |

Version History

(All revisions made to this document must be listed in chronological order. All revisions must be approved. Review and Approval can be done by an internal source or by the customer)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date of Revision | Description | Author | Reviewed By | Approved By |
| 1.0 | 03-08-2017 | Intial Draft | Balaji Ankalle | Jiten Pansara | Mahesh Keshatwar |
| 1.1 | **029-08-2017** | **Document Completed** | Balaji Ankalle | **Jiten Pansara** | **Mahesh Keshatwar** |
| 1.2 | **16-12-2018** | **Updated the content** | **Sreya Puthukudy** | **Jiten Pansara** | **Santosh Badiger** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Oracle: Tablespace Read only Conversion Issue for XTTS Migration

**Problem Description:**

As a part of the ATLAS replatforming project; there was a need to use Transportable tablespace method for the Migration from AIX to Linux Platform.

For the same; tablespace needs to be converted to read only mode during the XTTS migration window.

For the production database AMT1P of size 10TB; DBA team was unable to put the database into read only mode erroing out as below :

ORA-1640 signalled during: alter tablespace users read only...

**Impact:** Without the issue resolution; it was impossible to move ahead with AMT1P database Migration to LINUX platform as the XTTS Migration was the only feasible option for this migration considering the database size and cross platform migration requirement.

**Vendors Involved :**

Oracle Support

**Detailed Analysis and Steps involved for Resolution** :

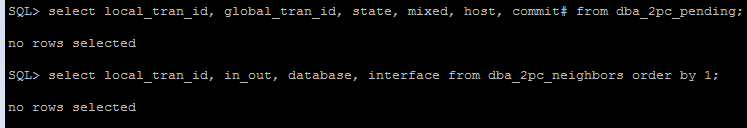
Durign the read only conversion for the tablespace

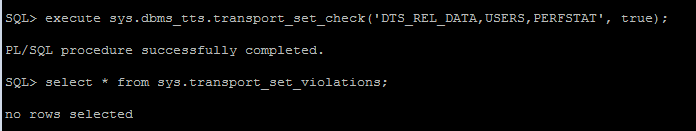
1) No sessions connected to database and all interfaces and application jobs are disabled.

2) No pending transactions in the database.

3) Tried with database in restricted mode and still fails with same error

4) There is no violations seen on sys.transport\_set\_violation





This was happening for even the newly created tablespaces. When attempted for a newly created database of same version on the same sever; it was not giving any error indicating something wrong within the database itself. (Pending Transaction – Distributed Transaction)

After a detailed discussion; DBA team decided to create a new undo tablespace and attempt this operation and identified that we were unable to drop the old Undo tablespace during pending offline rollback segements relating to the old Undo tablespace.

SQL> SELECT a.name,b.status FROM v$rollname a,v$rollstat b WHERE a.usn = b.usn AND a.name IN (SELECT segment\_name FROM dba\_segments WHERE tablespace\_name = 'UNDOTBS1');

NAME STATUS

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

\_SYSSMU1$ PENDING OFFLINE

\_SYSSMU2$ PENDING OFFLINE

\_SYSSMU5$ PENDING OFFLINE

\_SYSSMU10$ PENDING OFFLINE

A Case was raised with Oracle Support to cleanup the Distributed transaction as explained in the document ID : **835944.1.** PFB the output for the queries related to the document.

SQL> SELECT KTUXEUSN, KTUXESLT, KTUXESQN, /\* Transaction ID \*/ KTUXESTA Status,KTUXECFL Flags

FROM x$ktuxe WHERE ktuxesta!='INACTIVE'; 2

KTUXEUSN KTUXESLT KTUXESQN STATUS FLAGS

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

1 18 85277 PREPARED SCO|COL|REV|DEAD|EXTDTX

2 22 130991 PREPARED SCO|COL|REV|DEAD|EXTDTX

5 24 82474 PREPARED SCO|COL|REV|DEAD|EXTDTX

10 38 99382 PREPARED SCO|COL|REV|DEAD|EXTDTX

11 42 100147 PREPARED SCO|COL|REV|DEAD|EXTDTX

12 26 7456 PREPARED SCO|COL|REV|DEAD|EXTDTX

13 14 6009 PREPARED SCO|COL|REV|DEAD|EXTDTX

14 28 61568 PREPARED SCO|COL|REV|DEAD|EXTDTX

16 2 59566 PREPARED SCO|COL|REV|DEAD|EXTDTX

18 35 56733 PREPARED SCO|COL|REV|DEAD|EXTDTX

23 29 62698 PREPARED SCO|COL|REV|DEAD|EXTDTX

32 18 9172 PREPARED SCO|COL|REV|DEAD|EXTDTX

SQL> select local\_tran\_id, state from dba\_2pc\_pending;

no rows selected

Oracle provided a step by step approach to clear the pending transactions stated above.

Below prequisteies to be completed prior to the execution of these steps :

* Stop all application services and jobs running on the database
* Ensure that there are no other users connected in the database
* Full back up of the database is taken prior to carry out the activities
* Please spool all the execution and its outputs to a file, so that we will have clarity on the steps followed

1. Disable the distributed recovery

SQL> alter system disable distributed recovery;

System altered.

2. Insert the dummy record into pending\_trans$ for the dead transactions id

SQL> insert into pending\_trans$ (

LOCAL\_TRAN\_ID,

GLOBAL\_TRAN\_FMT,

GLOBAL\_ORACLE\_ID,

STATE,

STATUS,

SESSION\_VECTOR,

RECO\_VECTOR,

TYPE#,

FAIL\_TIME,

RECO\_TIME)

values( '1.18.85277',

306206,

'XXXXXXX.12345.1.2.3', /\* These values can be used without any \*/ 'prepared','P', /\* modification. Most of the values are \*/ hextoraw( '00000001' ), /\* constant. \*/ hextoraw( '00000000' ), /\* \*/ 0, sysdate, sysdate );

1 row created.

3. Insert the dummy record into pending\_sessions$ table for the dead transactions id

SQL> insert into pending\_sessions$ values( '1.18.85277', 1, hextoraw('05004F003A1500000104'), 'C', 0, 30258592, '', 146 );

1 row created.

4. Commit

5. Execute the force commit for the dead transaction id

SQL> commit force '1.18.85277';

Commit complete.

6 . If step 5 fails then delete the dummy entries inserted in step 2 & 3 or continue with steps

7. Commit

8. Enable the distributed recovery

SQL> alter system enable distributed recovery;

System altered.

9. Commit

10. Set the hidden parameter "\_smu\_debug\_mode" to 4

SQL> alter system set "\_smu\_debug\_mode" = 4;

System altered.

11. Commit

12. Execute the dbms\_transaction.purge\_lost\_db\_entry for the dead transaction id

SQL> exec dbms\_transaction.purge\_lost\_db\_entry( '1.18.85277' );

PL/SQL procedure successfully completed.

13. Commit

14. Set the hidden parameter "\_smu\_debug\_mode" back to 0

SQL> alter system set "\_smu\_debug\_mode" = 0;

System altered.

15. Commit

**Lessons Learned :**

For any issues with read only conversion for a tablespace; please check any pending distributed transactions as advised in the document ID : **835944.1** and immediately raise a case with Oracle Support as there can be different scenarios for the pending transactions and it is of utmost importance to follow the instructions as advised by advanced technical team from Oracle Support.