HUNTING DOWN UNDO ISSUES WITH ORACLE: A PRACTICAL APPROACH

Kamili Srinivas[1] Tanweer Khan[2] Ganti Siva[3]

1 Database Administrator, Middleware Engineering, ADP Inc, India.
2 Senior Project Leader, Middleware Engineering, ADP Inc, India.
3 Senior Consultant, Middleware Engineering, ADP Inc, India.

Abstract: This article discusses the challenges in managing UNDO data and several practical approaches one might follow while analyzing and resolving UNDO issues. The most frequent UNDO error: ORA-01555 snapshot-too-old error is dealt in detail and the practical resolution which is presented might help in resolving most of the problems while managing UNDO information.
KEYWORDS: Undo data; ORA-01555 snapshot too old error; Undo-retention; Undo tablespace.

I. INTRODUCTION

Whenever a transaction makes changes at the database level, Oracle copies the original data before modification, this original data is called UNDO DATA and it is stored in some special structures provided by Oracle called UNDO RECORDS. Oracle's undo records are stored in the UNDO TABLESPACE specified at the time of database creation.
Undo data is used for the following purposes:

- Rolling back transactions. A rollback operation can be the result of a user who wants to undo the changes of unintentional transaction, or it can be part of a recovery operation. Rollback is done using ROLLBACK statement.
- Providing read consistency for the SQL queries. This means a user can get consistent view of the data even though some other uncommitted changes are being performed against the data [1].

For instance, a transaction may update the value of an employee salary from 20,000 to 30,000. While this transaction is active and not committed, if some other query checks for the employee salary then the value 20,000 is provided. This is read consistency and the information comes from undo data.

Undo data remains in the undo tablespace even after a database shutdown. This makes Oracle's
undo management valuable for activities beyond rolling back transactions and providing read consistency, which are:

- Recovering terminated transactions.
- Analyzing older data by using Flashback Query.
- Recovering from logical corruptions using the Flashback features [2].

In the present paper, several practical approaches for managing the undo issues are presented. The issues are analyzed and appropriate practical resolutions are provided. In particular, the ORA-01555 snapshot too old error is dealt in detail.

II. ANALYSIS

Proper undo management means that necessary undo information is not overwritten by newer undo data. By setting the appropriate size for UNDO tablespace and the UNDO_RETENTION parameter there is an increase in chance that long-running queries can complete without receiving snapshot too old error [1]. Oracle 9i release introduced Automatic Undo management (AUM) which simplifies undo management to an extent as complexities in manually managing rollback segments is eliminated. AUM takes the entire responsibility of sizing and allocation of undo segments [4]. Setting the initialization parameter UNDO_MANAGEMENT to AUTO enables AUM mode.
If the database uses more than one undo tablespace, we have to specify which one to be used at the instance startup. This is done by specifying UNDO_TABLESPACE initialization parameter.
A dynamic parameter UNDO_RETENTION specifies the minimum length of time to retain undo data. The default value is 900 seconds. When in AUM mode the database will ignore any manual manual undo management mode SQL statements instead of returning error messages when the parameter UNDO_SUPPRESS_ERRORS is set TRUE. All the above mentioned parameters can be viewed in the snapshot provided as Fig 1. The RETENTION GAURENTEE clause while creating undo tablespace will guarantee the success of
queries at the price of compromising the success of DML operations.

```
SQL> sho parameter undo

NAME                                 TYPE                           VALUE
------------------------------------ ------------------------------ ------------------------------
undo_management                      string                         AUTO
undo_retention                       integer                        900
undo_suppress_errors                 boolean                        TRUE
undo_tablespace                      string                         UNDO
```

Fig 1. Snapshot of undo parameter list

```
$ ls -rlt|tail -3
-rw-r--r--   1 oracle   dba      240335363 Oct 12 06:17 dbH27Q_FULL_20101012_0100.dmp.Z
-rw-r--r--   1 oracle   dba         125798 Oct 12 06:17 dbH27Q_FULL_20101012_0100.log
-rw-r--r--   1 oracle   dba         127872 Oct 12 06:37 export_H27Q_20101012_0100.log
$ tail -5 dbH27Q_FULL_20101012_0100.log
. exporting indextypes
. exporting bitmap, functional and extensible indexes
EXP-00008: ORACLE error 1555 encountered
ORA-01555: snapshot too old: rollback segment number 1 with name "_SYSSMU1$" too small
EXP-00000: Export terminated unsuccessfully
/orabackup/H27Q[H27Q]$ oerr ora 1555
01555, 00000, "snapshot too old: rollback segment number %s with name \"%s\" too small"
// *Cause: rollback records needed by a reader for consistent read are
//         overwritten by other writers
// *Action: If in Automatic Undo Management mode, increase undo_retention
//          setting. Otherwise, use larger rollback segments
```

Fig 2. Export of data failed due to ORA-01555 error.

Now, the error ORA-01555 snapshot too old error is analyzed considering a live example. In the fig 2 an export of the database failed due to the error. It failed due to the rollback segment number _SYSSMU1 being too small. Hence the export of the database terminates unsuccessfully.

III. RESOLUTION

The immediate resolution would be increasing the UNDO_RETENTION parameter, but there is again an issue with increasing it to our will. There must be enough UNDO tablespace to support the increase in the undo_retention value.

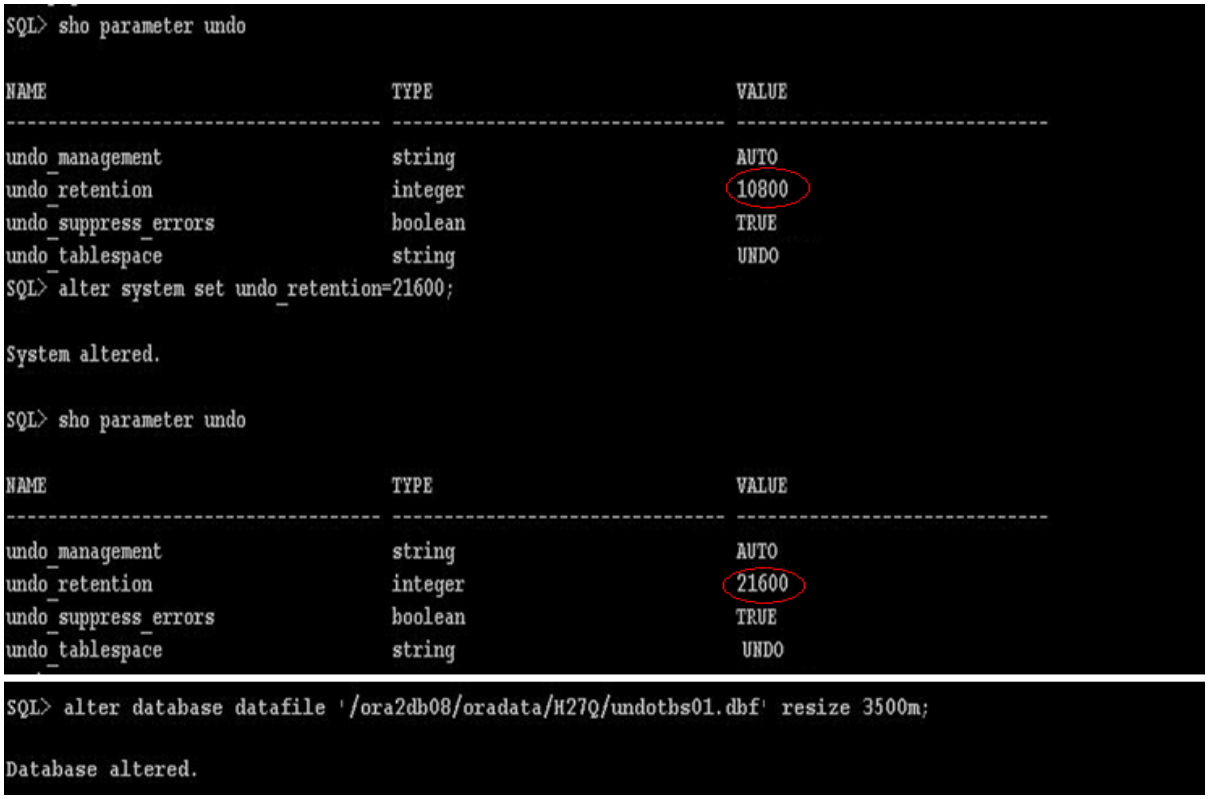

Fig 3. Increasing UNDO_RETENTION and UNDO Tablespace

From fig 2 it is evident that the export started at 1am and ended at 6:17am, so it ran more than 5hrs. Therefore, the UNDO_RETENTION should be increased to a value more than 5hrs. But the UNDO tablespace must be large enough to honor the increase in retention value. As we see in the fig3 the undo retention value is increased to a value greater that 5 hrs (21,600sec=6hrs>5hrs). Also, the undo tablespace has been increased to 3.5GB; this is done using ALTER DATABASE command. After the above changes were made the export ran successfully to completion.

It is advisable to make the production database's UNDO tablespace as autoextend ON, but not for the non-production databases. Moreover, there is a misconception that if the UNDO free space is not left we have to increase UNDO space. Though the undo space left may be zero, but unless there are expired segments left unused, there is no need to increase the undo space. The presence or absence of expired segments information is present in DBA_UNDO_EXTENTS. The listing 1 shows the expired segments present in the database from which the stats were extracted.
The dynamic performance views v$UNDOSTAT and DBA_UNDO_EXTENTS provide good statistics for managing UNDO issues. The Undo Advisor is also used to set new threshold for UNDO_RETENTION [5].

v$UNDOSTAT: The parameter UNXPSTEALCNT states the number of attempts to obtain undo space by stealing unexpired extents from other transactions. The parameter UNXPBLKRELCNT states the number of unexpired blocks removed from certain undo segments so they can be used by other transactions [2]. The parameter UNXPBLKREUCNT states then number of unexpired undo blocks reused by transactions. The listing 2 shows that UNXPSTEALCNT is 7, which means the undo space for the database is not enough and attempts were made to steal unexpired segments. This also means that some transactions have been failed. So, there is a need to increase the undo space. The parameters UNXPBLKRELCNT and UNXPBLKREUCNT show values greater than 0. Hence, some unexpired undo segments were overwritten which alerts for increase in undo space.

SQL> SELECT SEGMENT_NAME,BLOCK_ID,BYTES,STATUS FROM DBA_UNDO_EXTENTS;

SEGMENT_NAME	BLOCK_ID	BYTES	STATUS
_SYSSMU2$	1121	65536	EXPIRED
_SYSSMU2$	905	1048576	EXPIRED
_SYSSMU1$	10	57344	EXPIRED
_SYSSMU1$	33	65536	EXPIRED
_SYSSMU1$	1417	1048576	EXPIRED
_SYSSMU1$	2057	1048576	EXPIRED
_SYSSMU1$	1801	1048576	EXPIRED
_SYSSMU1$	649	1048576	EXPIRED
_SYSSMU1$	1289	1048576	EXPIRED

Listing 1: DBA_UNDO_EXTENTS information

SQL>SELECT TO_CHAR(BEGIN_TIME,'MM/DD/YYYY') BEGIN_TIME, TO_CHAR(END_TIME, 'MM/DD/YYYY') END_TIME ,UNXPSTEALCNT,UNDOBLKS,SSOLDERRCNT,EXPBLKRELCNT FROM v$UNDOSTAT 2 WHERE TO_CHAR(BEGIN_TIME , 'MM/DD/YYYY')='10/12/2010' AND ROWNUM<=10;

BEGIN_TIME	END_TIME	UNXPSTEALCNT	UNDOBLKS	UNXPBLKRELCNT	UNXPBLKREUCNT
10/12/2010	10/12/2010	7	1	5	3
10/12/2010	10/12/2010	4	0	2	0
10/12/2010	10/12/2010	0	0	0	4
10/12/2010	10/12/2010	2	2	1	1
10/12/2010	10/12/2010	0	2	0	0
10/12/2010	10/12/2010	0	85	0	0
10/12/2010	10/12/2010	0	17	0	0
10/12/2010	10/12/2010	0	14	0	0

Listing 2: v$UNDOSTAT information.

IV. CONCLUSION

In this paper, a practical approach to deal with undo errors is stated, which considers checking performance views like v$UNDOSTAT and DBA_UNDO_EXTENTS while changing any parameter values. The main objective of any database design is to provide data accessibility, data consistency and data maintenance. Resolving the UNDO issues will definitely help in fulfilling the above stated objectives.

V. REFERENCES

[1] Oracle Database Reference, 10*g* Release 2 (10.2) B14237-04 Copyright © 2002, 2009, Oracle. All rights reserved.Primary Author: Kathy Rich
[2] Expert Oracle database 11G Administration, Sam R. Alapati.
[3] Oracle Database Administrator's Guide, 10*g* Release 1 (10.1) Part No. B10739-01 Copyright © 2001, 2003 Oracle. All rights reserved. Primary Author: Ruth Baylis Contributing Authors: Paul Lane, Diana Lorentz.
[4] The Self-Managing Database: Deploying Oracle Database 11*g* in Embedded Environments.
[5] Oracle Database 2 Day DBA, 10*g* Release 1 (10.1) Part No. B10742-03 Copyright © 2004, Oracle. All rights reserved.Contributing Authors: Colin McGregor, Ruth Baylis, Sushil Kumar, Antonio Romero, David Austin, and Michele Cyran