MANAGING UNDO SPACE IN ORACLE9I

Megh Thakkar, Quest Software Inc.

INTRODUCTION

Rollback segments store the "before" image of data—the data as it existed prior to the beginning of a transaction. You need to have enough rollback segments in your database to support all of the concurrent transactions. When managing rollback segments, you need to manage their number and their space allocations.

If you have too few rollback segments, new transactions attempting to write data to the rollback segments will be forced to wait temporarily. You can use the scripts in this section to monitor the rollback header waits and determine if you have enough rollback segments in your database.

If the rollback segments are too small, then transactions may fail. If a large transaction requires more space than the rollback segment has already allocated, the rollback segment will extend into the free space remaining in the tablespace. If the rollback segment requires more space than is available in the tablespace, then the transaction causing the rollback segment to extend will fail. You can monitor the number of times the rollback segments had to extend to support the database transaction load.

If the extents within the rollback segment are too small, then the transaction will "wrap" from one extent to another within the rollback segment. Ideally, a transaction's rollback segment entry will fit entirely within one extent of the rollback segment, thereby minimizing the performance and internal space management issues associated with wraps. You can monitor the number of times the rollback segment entries wrap from one extent to another within the rollback segment.

MANAGING ROLLBACK SEGMENT CONTENTION

One or more transactions can use the same rollback segment. The result is efficient space utilization but also an increase in the potential for rollback segment contention. Table 1 below shows the various classes of blocks that provide information about rollback segments.

Table 1: Classes of Blocks Showing Rollback Segment Information

Block Class	Description
System undo header	Header blocks of the SYSTEM rollback segment
System undo block	Blocks of the SYSTEM rollback segment other than header blocks
Undo header	Header blocks of rollback segments other than the SYSTEM rollback segment
Undo block	Blocks other than header blocks of the rollback segments, other than the SYSTEM rollback segment

The following queries are useful in determining rollback segment contention:

- Waits per block class:

Input:

```
SELECT class, count
FROM V$WAITSTAT
WHERE class LIKE '%undo%'
AND COUNT > 0;
```

Output:

```
CLASS                COUNT
-----------------    ----------
system undo header   8206
system undo block     201
undo header          6731
undo block            320
```

- Total number of data requests:
-

Input:

```
SELECT SUM(value)  "DATA REQUESTS"
FROM V$SYSSTAT
WHERE name IN ('db block gets', 'consistent gets');
```

Output:

```
DATA REQUESTS
------------------------
         520975
```

Based on the preceding output, the percentage of waits for the rollback segment block classes can be calculated as follows:

- Wait for system undo header is (8206/520975)*100 = 1.575%
- Wait for system undo block is (201/520975)*100 = 0.038%
- Wait for undo header is (6731/520975)*100 = 1.29%
- Wait for undo block is (320/520975)*100 = 0.061%
-

There is contention for rollback segments because the waits for system undo header blocks and the waits for undo header blocks are greater than 1% of the total requests.

Contention for a block class is indicated by waits for that block class being greater than 1% of the total block requests.

You can use several techniques to reduce the contention for rollback segments:

- Increase the number of rollback segments.
- Set the storage parameter NEXT to the same value as INITIAL for rollback segments.
- Set the storage parameter MINEXTENTS to at least 20.
- Set the storage parameter OPTIMAL to INITIAL * MINEXTENTS.

- Set the storage parameter INITIAL to at least the number of undos generated by the transactions. You can determine the number of undos generated by transactions as follows:

```
SELECT MAX(USED_UBLK)
FROM v$transaction;
```

- Ensure that you have plenty of free space in the rollback tablespace.
- Set the initial number of rollback segments to be $N/4$, if you have a good estimate of the number of concurrent transactions, say N.

SYSTEM MANAGED UNDO

In Oracle versions prior to Oracle9i, the undo space is managed with the help of rollback segments. Oracle9i databases are capable of managing their own undo (rollback) space. The database administrators don't have to worry about planning and tuning the number and size of rollback segments or determine a strategy of assigning transactions to appropriate rollback segments. In addition, Oracle9i allows administrators to allocate their undo space in a single undo tablespace. This self-managing of undo space allows Oracle to take care of issues such as undo block contention, consistent read retention and utilization of undo space; normally a database administrator has to spend a considerable amount of time in working through these issues.

It should be noted, however, that if you desire you can still make use of rollback segments to manage undo space in an Oracle9i database. The UNDO_MANAGEMENT initialization parameter determines the method of managing undo space in use. If you use the rollback segment method for undo space management then you are said to be using rollback segment undo (RBU) scheme and if you use the undo tablespace method for managing of undo space then you are said to be using the system-management undo (SMU) scheme.

The following initialization parameters control the management of undo space in an Oracle9i database:

- UNDO_MANAGEMENT

If set to 'auto', SMU scheme is in effect and if set to 'manual', RMU scheme is in effect. The default for Oracle9i is 'auto'.

- UNDO_TABLESPACE

This is dynamically changeable parameter that specifies the undo tablespace to use in an SMU-managed Oracle9i database. When UNDO_MANAGEMENT is set to 'auto' and the instance is started, the undo tablespace specified by UNDO_TABLESPACE is used. This undo tablespace should already have been created. If this parameter is omitted then the first available undo tablespace in the database is chosen and if none is available then the SYSTEM rollback segment is used.

- UNDO_RETENTION

This is a dynamically changeable parameter that specifies the length of time to retain undo. The default is 5 minutes.

- UNDO_SUPRESS_ERROR

This is a dynamically changeable parameter that specifies whether or not error messages are to be generated if RMU SQL statements are issued while running the database in the SMU-mode. Default value of 'false' indicates that errors message are not to be supressed.

- ROLLBACK_SEGMENTS

When operating in the RMU mode, this parameter specifies the rollback segments to acquire at start up

- TRANSACTIONS

When operating in the RMU mode, this parameter specifies the maximum number of concurrent transactions

- TRANSACTIONS_PER_ROLLBACK_SEGMENT

When operating in the RMU mode, this parameter specifies the maximum number of concurrent transactions that each rollback segment is expected to handle

- MAX_ROLLBACK_SEGMENTS

When operating in the RMU mode, this parameter specifies the maximum number of rollback segments that can be online for this instance

The V$UNDOSTAT view can be used to monitor the effects of transaction execution on undo space in the current instance.

The following script shows how to obtain statistics about undo space consumption in the instance.

```
select TO_CHAR(MIN(Begin_Time),'DD-MON-YYYY HH24:MI:SS')
                               "Begin Time",
       TO_CHAR(MAX(End_Time),'DD-MON-YYYY HH24:MI:SS')
                               "End Time",
       SUM(Undoblks)       "Total Undo Blocks Used",
       SUM(Txncount)       "Total Num Trans Exec",
       MAX(Maxquerylen)    "Longest Query(in secs)",
       MAX(Maxconcurrency) "Highest Concurrent Trans Count",
       SUM(Ssolderrcnt),
       SUM(Nospaceerrcnt)
from V$UNDOSTAT;
set feedback 6 pagesize 24 term on
```

Sample output of running the above script is shown below.

```
Begin Time           End Time             Total Undo Blocks Used
-------------------- -------------------- ----------------------
Total Num Trans Exec Longest Query(in secs) Highest Concurrent Trans Count
-------------------- ---------------------- ----------------------------
SUM(SSOLDERRCNT) SUM(NOSPACEERRCNT)
---------------- ------------------
11-JUL-2001 12:01:57 11-JUL-2001 12:28:12                     20
                  30                     11                              4
               0                  0
```

ABOUT THE AUTHOR:

Megh Thakkar is the Director of Database Technologies at Quest Software. Previously, he worked as a technical specialist at Oracle Corporation. He holds a master's degree in computer science and a bachelor's degree in electronics engineering. Megh also holds several industry vendor certifications, including OCP, MCSE, Novell Certified ECNE, SCO UNIX ACE, and he is a Lotus Notes Certified Consultant. Voted as the best speaker at the NSWOUG'2000 in Sydney and at the VOUG'2001 in Melbourne, Australia; he is a frequent presenter at OracleWorld and various international Oracle User Groups.

Megh is the author *of Oracle9i: Instant PL/SQL scripts*, *e-Business for the Oracle DBA*, *E-commerce Applications Using Oracle8i and Java from Scratch* and *Teach Yourself Oracle8i on Windows NT in 24 Hours*. He has also co-authored several books, such as *Migrating to Oracle8i*, *Special Edition Using Oracle8/8i*, *Oracle8 Server Unleashed*, *C++ Unleashed*, *COBOL Unleashed*, *Oracle Certified DBA* and, *Using Oracle8*. Megh is a renowned Oracle specialist who has performed Oracle development, consulting, support, and DBA functions worldwide over the past thirteen years.

REFERENCES:

1. "Oracle9i: Instant PL/SQL Scripts" by Kevin Loney, Megh Thakkar and Rachel Carmichael (McGraw-Hill Oracle Press; ISBN: 0072132183)
2. "e-Business for the Oracle DBA" by Megh Thakkar (Sams; ISBN: 0672321475)
3. "e-Commerce Applications using Oracle8i and Java from Scratch" by Megh Thakkar (Que; ISBN: 0789723387)

MEGHRAJ T. THAKKAR

DIRECTOR OF DATABASE TECHNOLOGIES

QUEST SOFTWARE

MEGH.THAKKAR@OZ.QUEST.COM