**ORA-01555错误模拟详解**

下面是摘自联机文档Database Error Messages关于ORA-01555的描述

ORA-01555: snapshot too old: rollback segment number string with name "string"

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

出现ORA-01555错误，通常有2种情况：

（1）SQL语句执行时间太长，或者UNDO表空间过小，或者事务量过大，或者过于频繁的提交，导致执行SQL过程中进行一致性读时，SQL执行后修改的前镜像（即UNDO数据）在UNDO表空间中已经被覆盖，不能构造一致性读块（CR blocks），这种情况最多。

（2）SQL语句执行过程中，访问到的块，在进行延迟块清除时，不能确定该块的事务提交时间与SQL执行开始时间的先后次序。 这种情况很少。

 第1种情况解决的办法：

（1）增加UNDO表空间大小

（2）增加undo\_retention 时间，默认只有15分钟

（3）优化出错的SQL，减少查询的时间，首选方法

（4）避免频繁的提交

通过下面的例子，来看看ORA-01555是如何产生的

1.确定系统当前系统的undo的相关信息

SQL> show parameter undo

NAME                                 TYPE        VALUE

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

undo\_management                      string      AUTO

undo\_retention                       integer     900

undo\_tablespace                      string      UNDOTBS1

SELECT tablespace\_name,BYTES/1024/1024,file\_name from dba\_data\_files WHERE tablespace\_name LIKE 'UNDO%';

SQL> col file\_name for a32;

SQL> /

FILE\_NAME                 BYTES/1024/1024 TABLESPACE\_NAME

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

+DATA/orcl/datafile/undotbs1.dbf        140 UNDOTBS1

2. 重新创建一个大小为1m undo表空间，并且切换到这个undo表空间

SQL> create undo tablespace undotbs2 datafile '+DATA/orcl/datafile/undotbs2.dbf' size 1m;

Tablespace created.

SQL> alter system set undo\_tablespace=undotbs2;

System altered.

SQL> show parameter undo;

NAME                             TYPE        VALUE

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

undo\_management                      string      AUTO

undo\_retention                       integer     900

undo\_tablespace                      string      UNDOTBS2

SQL>

3.创建一个测试表test,当我们往表里插入数据的时候，就报了下面undotbs2无法扩展的错误，我们知道insert语句会写回滚段，由于这个insert的数据比较大，导致undo迅速增长，当达到1m的时候，oracle会首先检查有没有undo块可以覆盖，刚好都是没有commit的数据块，所以这时候会试着扩展回滚段，前面创建没有指定自动扩展，因此就产生了ORA-30036的错误

SQL> create table test as select object\_id,object\_name from dba\_objects;

Table created.

SQL> insert into test select \* FROM TEST;

insert into test select \* FROM TEST

           \*

ERROR at line 1:

ORA-30036: unable to extend segment by 8 in undo tablespace 'UNDOTBS2'

SQL> select segment\_name,bytes/1024/1024 from dba\_segments where segment\_name='TEST';

SEGMENT\_NAME BYTES/1024/1024

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

TEST 5

4.清除这个表，重新创建一个表，插入两行数据

SQL>drop table test;

SQL> create table test (id int,name varchar2(10));

Table created.

SQL> insert into test values(1,'a');

1 row created.

SQL> insert into test values(2,'b');

1 row created.

SQL> commit;

Commit complete.

5.这里通过游标变量来查询这个表，这里只是把这个表的查询结果集赋给了类型为refcuersor的变量i，如果要输出变量的值用print i。假设这里时刻为t1

SQL> variable i refcursor;

SQL> exec open :i for select \* from test;

PL/SQL procedure successfully completed.

6. 打开另外一个session，修改这个表并做提交，这里通过一个死循环，反复修改并且提交

修改之前的数据，这样做的目的主要是根据undo的特性覆写undo块

SQL> select \*from test;

       ID NAME

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

        1 a

        2 b

SQL>    declare

 2       i number := 1;

 3      begin

 4       loop

 5         update test set id=i where name = 'b';

 6        i := i+1;

 7         if mod(i,100)=0 then

 8           commit;

 9         end if;

10     end loop;

11    end;

12  /

7. 等待10秒左右输出另外一个session的游标变量i，这个时候比较著名的ORA-01555的错误就出现了。

SQL> print i

ERROR:

ORA-01555: snapshot too old: rollback segment number 13 with name

"\_SYSSMU13\_3721975596$" too small

no rows selected

 导致这个错误的原因就是sql查询时间太长。我们通过游标变量来制造一个长时间的查询。这里并没有提示undo段无法扩展的错误，这是为什么，根据前面的2个例子的结果分析，就可以推出undo段提交后的数据块是可以被覆盖的，这里undo虽然只有1m但是我们update产生的undo数据量并为达到1m，由于这个循环产生了大量的undo，所以在达到1m的时候发现无法扩展，就会覆写commit后的undo块，因此undo表空间过小也是导致这一错误的原因，根据Cause: rollback records needed by a reader for consistent read are overwritten by other writers这个原因，当用户发出输出i的请求时会在buffer cache中构造cr块，然后返回给用户，cr块就是从修改前的undo块构造的，这里undo块已经被覆盖了无数遍了，所以就报了ORA-01555的错误。

8.前面的insert无法直观的分析出commit前的数据是否能够被覆盖，把循环中的commit去掉。先猜想下，这里会报错什么样的错误，如果上面分析正确，毫无疑问会报出ORA-30036的错误，来验证一下

SQL>    declare

 2       i number := 1;

 3      begin

 4       loop

 5         update test set id=i where name = 'b';

 6        if mod(i,100)=0 then

 7           i := i+1;

 8         end if;

 9     end loop;

10    end;

11  /

  declare

\*

ERROR at line 1:

ORA-30036: unable to extend segment by 8 in undo tablespace 'UNDOTBS2'

ORA-06512: at line 5

 联机文档给出了解决方法Action: If in Automatic Undo Management mode, increase undo\_retention setting. Otherwise, use larger rollback segments，所以前面的实验模拟只需要把undo表空间扩大或者自动增长开启就行了，这真实环境中往往就可能是sql语句查询时间过长，所以想解决问题还得从优化sql开始。

undo\_retention的作用，这个值默认是900s，意思就是说commit后的在undo段中保留900s，但并不保证在900s时间内一定不被覆写，从前面的实验就可以看出在900s内已经被覆写，如果要保证undo数据一定在900s不被覆写的话，需要设置undo段retention的值为guarantee

SQL>  select tablespace\_name,RETENTION from dba\_tablespaces where tablespace\_n

e='UNDOTBS2';

TABLESPACE\_NAME                RETENTION

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

UNDOTBS2                       NOGUARANTEE

1. 通过下面的命令修改为guarantee

SQL> alter tablespace undotbs2 retention guarantee;

Tablespace altered.

SQL>  select tablespace\_name,RETENTION from dba\_tablespaces where tablespace\_n

e='UNDOTBS2';

TABLESPACE\_NAME                RETENTION

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

UNDOTBS2                       GUARANTEE

2. 如果保证900s，我们在执行前面的语句，可想而知就会报ORA-30036,还是验证一下

SQL>    declare

 2       i number := 1;

 3      begin

 4       loop

 5         update test set id=i where name = 'b';

 6        i := i+1;

 7        if mod(i,100)=0 then

 8          commit;

 9         end if;

10     end loop;

11    end;

12  /

  declare

\*

ERROR at line 1:

ORA-30036: unable to extend segment by 8 in undo tablespace 'UNDOTBS2'

ORA-06512: at line 5

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

闪回删除的表

SQL> drop table test;

SQL> select \* from recyclebin;

SQL> flashback table test to before drop;

SQL> select \* from recyclebin;

SQL> drop table test purge;

SQL> select \* from recyclebin;

no rows selected