杀锁后有可能出现的UNDO回滚导致SMON导致行锁等待

如果杀掉的session正在运行比较大的事务，则kill会话后必然会出现长时间等待smon进程回滚undo，在smon进行回滚的过程中，相关行数据无法进行修改和删除等操作，因此在杀掉一个大事务session后，有可能会出现数据库后台进程会话SMON持锁的行锁等待。

smon进程是background的后台进程不能进行手动kill，否则实例会崩溃。而且即使在此时重启数据库，也无法取消回滚会话。

应该如何处理：

首先如果查到持锁会话是后台SMON进程，则应该检查是否存在死事务回滚。

SELECT KTUXEUSN,

       KTUXESLT,

       KTUXESQN, /\* Transaction ID \*/

       KTUXESTA Status,

       KTUXECFL Flags,KTUXESIZ

  FROM x$ktuxe

 WHERE ktuxesta != 'INACTIVE' and  KTUXECFL like '%DEAD%';

如果存在死事务回滚则使用如下sql评估回滚结束所需要的时间

 set serveroutput on

 declare

 l\_start number;

 l\_end number;

 begin

 select ktuxesiz into l\_start from x$ktuxe where KTUXEUSN=191 and KTUXESLT=19;

 dbms\_lock.sleep(60);

 select ktuxesiz into l\_end from x$ktuxe where KTUXEUSN=191 and KTUXESLT=19;

 dbms\_output.put\_line('time est Day:'|| round(l\_end/(l\_start -l\_end)/60/24,2));

 end;

 /

其中KTUXEUSN  and KTUXESLT 对应的值为上一条sql的查询结果。

单位是 天。

数据库默认回滚采用并行回滚即参数fast\_start\_parallel\_rollback为low

查询目前回滚参数fast\_start\_parallel\_rollback的值。

su - oracle

sqlplus / as sysdba

show parameter fast\_start\_parallel\_rollback

有时并行回滚会因为并行会话之间协调而导致回滚所需时间较长，如果以上查询结果较长可以通过修改fast\_start\_parallel\_rollback为false来采用串行回滚。

操作方法是：

How to Disable Parallel Transaction Recovery When Parallel Txn Recovery is Active (Doc ID 238507.1)

解决方案：

在线地把并行恢复改为串行。

1. 找到smon的 oracle pid （注意不是os pid）

SQL> select pid, program from v$process where program like '%SMON%';

       PID PROGRAM

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

         6 [oracle@stsun7](mailto:oracle@stsun7 \\t _blank) (SMON)

2. disable smon transaction cleanup #关闭事务回滚

SQL> oradebug setorapid 'SMON's Oracle PID';   --注意此处不带引号直接写PID即可

SQL> oradebug event 10513 trace name context forever, level 2

3.从os层面kill掉那些正在执行并行事务恢复的pq slave进程。可以通过V$FAST\_START\_SERVERS 来找到这些pq slave进程

select SPID from V$PROCESS where PID in (select PID from V$FAST\_START\_SERVERS);

然后从os层面kill 掉上面select语句的查询结果： kill -9 spid

若是cluster环境，以上1-3操作需要**在所有的实例**上修改

4. 关闭fast\_start\_parallel\_rollback

alter system set fast\_start\_parallel\_rollback=false;

5.重新启动 事务恢复（transaction recovery ）

SQL> oradebug setorapid 'SMON's Oracle PID';

SQL> oradebug event 10513 trace name context off

若是cluster环境，这两条命令需要在**所有的实例**上修改

过几分钟后再进行时间评估。如果时间变得更慢则改回原值

alter system set FAST\_START\_PARALLEL\_ROLLBACK = low;

同样如果改成串行回滚后，一旦回滚结束记得将FAST\_START\_PARALLEL\_ROLLBACK改回原值。

剩下的就只能等待了。

1. 常见行锁等待问题原因

1、select for update ，虽然是查询但是会持锁

2、update或者delete 某行数据，但是没有提交，也没回滚。

3、存在唯一约束或者主键列insert插入了一行数据，没有提交也没回滚，其他会话插入值时存在相同值。

其他原因。

1. 行锁问题发生后应如何定位：

可以通过ASH、或者AWR去判断，存在大量行锁等待的问题时段是否异常SQL.

可以确认历史锁事件，但同样无法精确定位持锁sql：

select t.SEQ#,

       (max(t.SAMPLE\_TIME) - min(t.SAMPLE\_TIME)) "持续时间",

       t.SESSION\_ID "当前会话id",

       t.SESSION\_SERIAL# "当前会话SERIAL#",

       t.event "等待事件",

       t.BLOCKING\_SESSION "阻塞会话id",

       t.BLOCKING\_SESSION\_SERIAL# "阻塞会话SERIAL#"

  from  dba\_hist\_active\_sess\_history  t

 where sample\_time between

       to\_date('2015-11-13 9:30:00', 'yyyy-mm-dd hh24:mi:ss') and

       to\_date('2015-11-13 10:45:00', 'yyyy-mm-dd hh24:mi:ss')

 group by t.SESSION\_ID,

          t.SESSION\_SERIAL#,

          t.BLOCKING\_SESSION,

          t.BLOCKING\_SESSION\_SERIAL#,

          t.EVENT,

          t.SEQ#