

在OLTP的环境下使用大事务出现的问题Oracle认证考试 PDF
转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/577/2021_2022__E5_9C_A8OLTP_E7_9A_84_c102_577864.htm 在应用环境中，常常需要保证几张表相关数据的一致性，为了应对这种需求，开发工程师常常会使用事务，把一些insert,update等语句绑在一起，形成一个事务(Transaction),比如如下的伪代码示例(事务1)

```
: begin transaction insert into messgae(id,col1,col2...)
values(#id,#col1,#col2...). update thread set
post_count=nvl(post_count,0) + 1 where id=#v_id. update test_user
set sum_reply = sum_reply + 1 where userid=#v_userid.end
transaction
```

开发工程师为了实现应用数据完整性的需要，把这几条SQL绑在一起形成一个大事务。在OLTP业务发展的早期阶段，这样设计并不会遇到任何问题，并可以最大程度的保证应用层数据的完整性。但随着OLTP业务量的发展，并发访问量的快速增长，这个大事务开始出现问题，阻塞现象异常严重，导致应用服务器页面响应时间明显加长，并已影响到业务的正常运行!经过调查发现，在系统中还存在这样一个更新SQL(事务2)：

```
update thread set
view_count=nvl(view_count,0) + 1 where id=#v_id.
```

而这个因外界访问量的增大，执行次数异常的高。很明显，当事务1执行时，有多个语句要执行，执行时间较长，如果此时有大量的事务2相要执行，我所指的是更新相同的id的记录。那么即会发生出现大量的enqueue等待。这是采用当前事务策略下，在OLTP环境下，我们不得不面对的问题. 在刚开始出现大量锁enqueue等待时，针对事务1,事务2执行频率的不同：事务1

低，事务2高。可以采用一定的策略来减少事务2的数据库执行次数，比如先将此更新放到缓存里，每隔一定的时间间隔更新到数据库里，减少事务1与事务2的碰撞机会。随着OLTP的进一步发展，并发访问量的进一步提高，因事务1本身的事务较大，事务1发生的频率也会越来越高，那时，事务1会与事务1本身产生阻塞，那时我们怎么解决？看来解除事务是唯一的方法，这里有另外一种观点，完全解除事务，即把事务1中的每一个语句都拆开，示例如下：

```
begin transaction insert into
messgae(id,col1,col2...) values(#id,#col1,#col2...).end transaction
begin transaction 0update thread set post_count=nvl(post_count,0)
1 where id=#v_id.end transactionbegin transaction 0update test_user
set sum_reply = sum_reply 1 where userid=#v_userid.end
transaction
```

这种方案可以看成事务1的极端，事务1的完全对立面，数据的一致性完全没有保障。如果数据库出现down机，或者应用服务器down掉，那么数据很有可能出现不一致的。为了减少发生数据不一致的概率，可以采用折衷的方案在一定程度上解除事务。对以上事务1的业务进行分析，发现了如下的规律：

```
insert into messgae(id,col1,col2...)
values(#id,#col1,#col2...).
```

这条SQL可以看成"私有"的，不会产生阻塞(这个表上不能有位图索引)

```
0update thread set
post_count=nvl(post_count,0) 1 where id=#v_id.
```

这条SQL是"公共"的，很容易发生多个session更新同一条记录，容易产生阻塞

```
0update test_user set sum_reply = sum_reply 1 where
userid=#v_userid.
```

这条SQL更新自己的东西，也可以看成"私有"的，不会产生阻塞

对以上作分析后，产生了另外一种方案：把"公共"的绑成一个事务，把"私有"的绑成一个事务，原来的

事务1变成2个事务，示例如下：100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com