MySQL间隙锁问题

## 概念

间隙锁（Gap Lock）：锁加在不存在的空闲空间，可以是两个索引记录之间，也可能是第一个索引记录之前或最后一个索引之后的空间。

最近用户反馈说系统老是出现insert时，等待超时了，最后发现是insert间隙锁！间隙锁是innodb中行锁的一种，但是这种锁锁住的却不止一行数据，他锁住的是多行，是一个数据范围。间隙锁的主要作用是为了防止出现幻读，但是它会把锁定范围扩大，有时候也会给我们带来麻烦，我们就遇到了。

在数据库参数中， 控制间隙锁的参数是：innodb\_locks\_unsafe\_for\_binlog，这个参数默认值是OFF， 也就是启用间隙锁， 他是一个bool值， 当值为true时表示禁用间隙锁。

那为了防止间隙锁是不是直接将innodb\_locaks\_unsafe\_for\_binlog设置为true就可以了呢？ 不一定！而且这个参数会影响到主从复制及灾难恢复， 这个方法还尚待商量。

## 问题原因

间隙锁的出现主要集中在同一个事务中先delete后，insert的情况下， 当我们通过一个参数去删除一条记录的时候， 如果参数在数据库中存在，那么这个时候产生的是普通行锁，锁住这个记录， 然后删除， 然后释放锁。如果这条记录不存在，问题就来了， 数据库会扫描索引，发现这个记录不存在， 这个时候的delete语句获取到的就是一个间隙锁，

然后数据库会向左扫描扫到第一个比给定参数小的值，向右扫描扫描到第一个比给定参数大的值， 然后以此为界，构建一个区间， 锁住整个区间内的数据， 一个特别容易出现死锁的间隙锁诞生了。

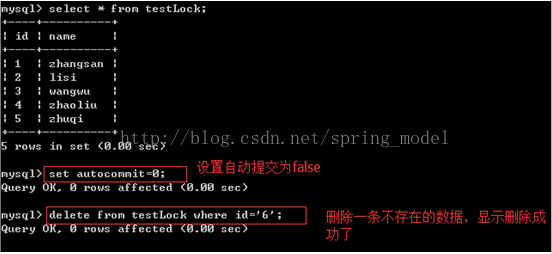
## 实例分析

举个例子：

表testLock,有两个属性id,和name.有如下数据。

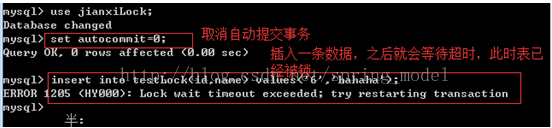


开启一个会话： session 1



sql> set autocommit=0;                 ##取消自动提交  
sql> delete from testLock where id = ‘6’;

再开启一个会话： session 2



sql> set autocommit=0;##取消自动提交

sql> insert into testLock(id,name) values(‘6’,’hahaha’);

在没有并发，或是极少并发的情况下， 这样会可能会正常执行，在Mysql中， 事务最终都是穿行执行， 但是在高并发的情况下， 执行的顺序就极有可能发生改变， 变成下面这个样子：

sql> delete from testLock where id = ‘6’;

sql> insert into testLock(id,name) values(‘6’, ‘hahaha’);

这个时候最后一条语句：insert into testLock(id,name) values(‘6’, ‘hahaha’); 执行时就会爆出死锁错误。因为删除id = 6这条记录的时候，

id为6之后的部分都被锁住了， 他们都取得了这一个数据段的共享锁， 所以在获取这个数据段的排它锁时出现死锁。

**解决办法**

前面说了， 通过修改数据库的参数innodb\_locaks\_unsafe\_for\_binlog来取消间隙锁从而达到避免，这种情况的死锁的方式尚待商量。那就只有修改代码逻辑， 存在才删除（有的说存在才删除也会出现间隙锁的问题，所以尽量不要采用先删除的办法。），尽量不去删除不存在的记录。一般采用：

* 更新的时候，比如将id为1的name为李四的更新为张三，可以使用如下sql语句：

update tab\_name set name=’张三’where id=1 and name=’李四’

* 需要insert的时候，可以先查询，如果存在，则直接更新，如果不存在，则insert。