MySQL死锁分析

当程序并发update同一张表，或者并发insert 一张表的时候都会出现数据库死锁问题。当并发insert一张表可能会出现间隙锁（小概率事件），这里先不谈，我主要描述update的常见的一种场景。

开启mysql死锁监控

create table InnoDB\_monitor(a INT) engine=InnoDB;

linux 环境重启myql

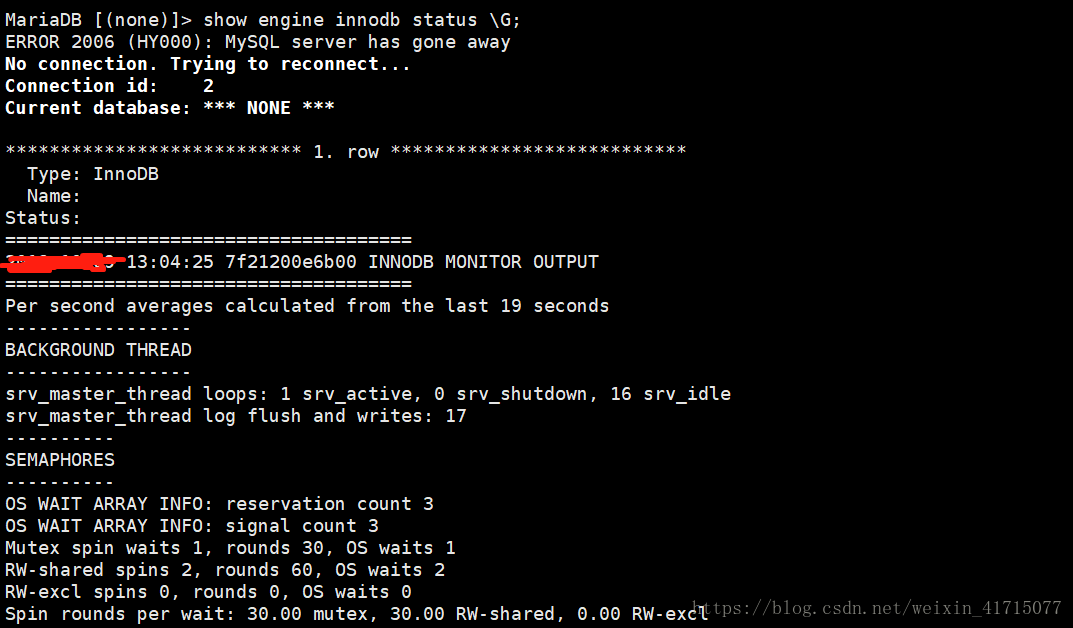
查死锁信息

查看死锁日志

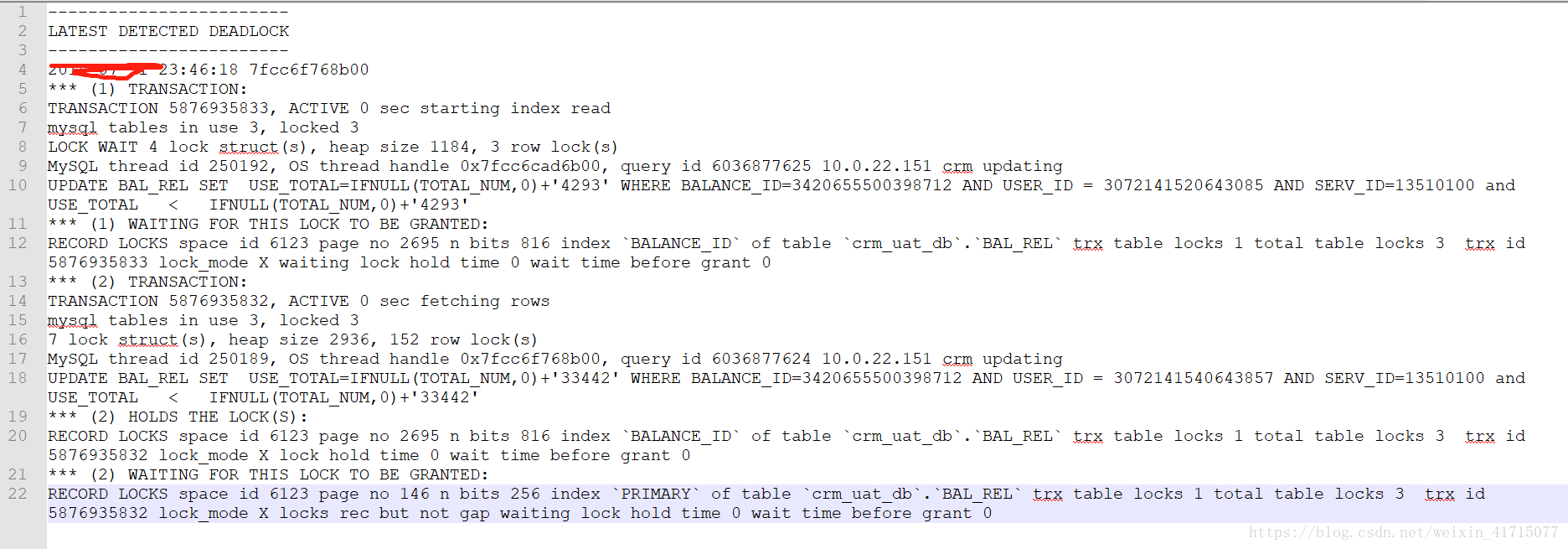
在linux系统中：

先用 root 账号登录 mysql -uroot -proot。

然后用命令查看死锁信息 show engine innodb status \G;



在日志中找 LATEST DETECTED DEADLOCK 字样的日志，就是死锁日志。下面是我整理并截取的部分日志



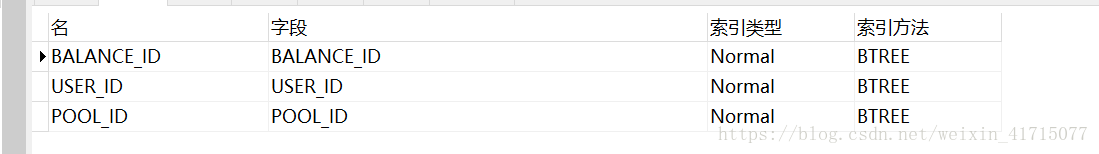
问题描述

可以看到出现死锁的两条SQL分别是：

UPDATE BAL\_REL SET USE\_TOTAL=IFNULL(TOTAL\_NUM,0)+‘4293’ WHERE BALANCE\_ID=3420655500398712 AND USER\_ID = 3072141520643085 AND SERV\_ID=13510100 and USE\_TOTAL < IFNULL(TOTAL\_NUM,0)+‘4293’

和

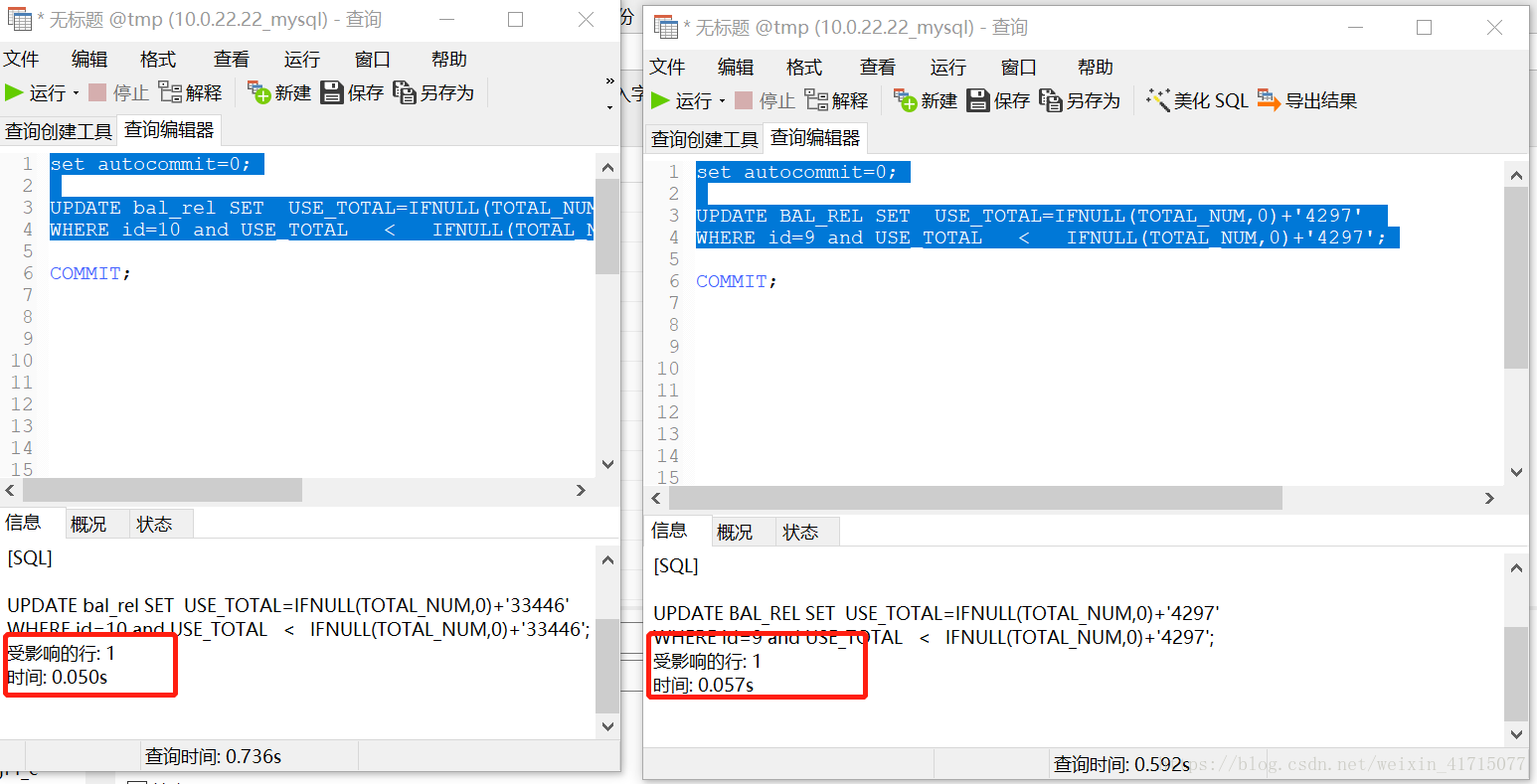
UPDATE BAL\_REL SET USE\_TOTAL=IFNULL(TOTAL\_NUM,0)+‘33442’ WHERE BALANCE\_ID=3420655500398712 AND USER\_ID = 3072141540643857 AND SERV\_ID=13510100 and USE\_TOTAL < IFNULL(TOTAL\_NUM,0)+‘33442’

看下表模型设计：  
由于表数据量比较大，加了三个索引，而且都是BTREE索引。  


这样对更新同一个BALANCE\_ID的update SQL 就有问题。由于BTREE存在，update 会锁住 BALANCE\_ID索引节点下 每一条记录，就不是行级锁了。

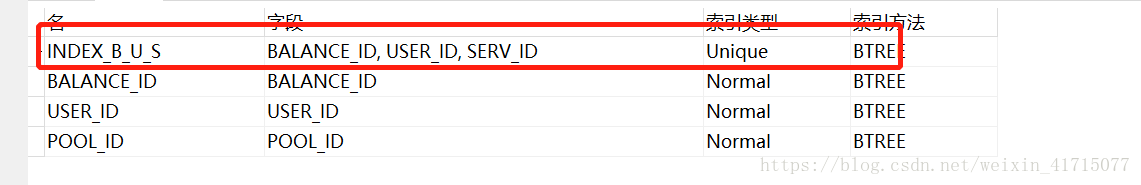
**问题解决方案**

优化表结构  
添加自增长的主键：  
通过主键ID ,update每条数据，这时候使用的都是行锁。  
设置手动提交事务 分别执行SQL，set autocommit=0; 并发执行两个SQL没有出现死锁。



但是这样**效率比较低**，需要先查询到主键，然后update

（2）优化索引  
可以看到SQL中where条件用了三个字段：BALANCE\_ID，USER\_ID，POOL\_ID  
添加一个组合索引 BALANCE\_ID，USER\_ID，POOL\_ID



执行下SQL，没有出现锁等待，说明这种方式也是可以行,也不需要查询主键ID，相对来说效率会高点，但是会增加索引磁盘空间，insert的效率也会降低。

