Atitit 死锁解除

目录

[1. 操作语句整理分析 1](#_Toc17104)

[2. 调配置（减少死锁概率） 1](#_Toc24994)

[2.1. - 调整隔离级别 和事务日志刷盘模式 1](#_Toc10617)

[2.2. 调整mysql内部线程和事务超时，超时释放 2](#_Toc22784)

[2.3. 配置调整线程池 立即释放 2](#_Toc6521)

[2.4. 增加死锁检测和自动释放 2](#_Toc32338)

[3. 较大的改动 2](#_Toc25255)

[3.1. 升级mysql8 2](#_Toc11260)

[3.2. 只迁移账户表user 》》pgsql 3](#_Toc8242)

[3.3. 分库分表user 3](#_Toc31067)

[4. 改代码模式 3](#_Toc6447)

可能死锁的表 User表，包括用户信息，账户金额。。 ，，还有个存取款表。。

# 操作语句整理分析

# 调配置（减少死锁概率）

## - 调整隔离级别 和事务日志刷盘模式

set global transaction isolation level read committed;

show global variables like '%tx\_isolation%';

set global innodb\_flush\_log\_at\_trx\_commit=2;

看他们对死锁和性能的提升程度。。理论上提升了处理速度也会降低死锁概率

## 调整mysql内部线程和事务超时，超时释放

超时类参数与死锁检测

set global innodb\_deadlock\_detect=1; -- innodb\_deadlock\_detect 1 is on open

show global variables like '%timeout%';

set global lock\_wait\_timeout = 30; -- dml lock wait

set global deadlock\_timeout\_long=30;

set global deadlock\_timeout\_short=30;

set global wait\_timeout=30; #trans exe time out

set global interactive\_timeout=30;

set global innodb\_lock\_wait\_timeout=30;

## 配置调整线程池 立即释放

## 增加死锁检测和自动释放

原理大概是查询元数据库information\_schema，mysql单条修改语句也会加单语句事务。。

得到所有的事务，加锁的事务，等待加锁的事务

所有对数据库连接。。。

对其进行循环判断，判断开始时间，如果时间大于一定时间，就将其强制释放

但可能对少数确实需要长时间类的sql语句有影响

# query trx

SELECT \* FROM information\_schema.INNODB\_TRX;

kill 12; SELECT \* FROM information\_schema.INNODB\_TRX;

INNODB\_LOCKS ，INNODB\_LOCKS\_wait

# 较大的改动

## 升级mysql8

优点：兼容性高，性能提升大约要比5.6 提升7倍。。

主要问题就是迁移数据量较多，

而且可能的行锁机制，还会导致死锁，但概率应该会降低

## 只迁移账户表user 》》pgsql

优点， ，性能也方便提升版本升级 ，独立升级表

（验证下他的行锁模式）

面临对user表的join查寻问题， 暂时搜索类下代码里面，貌似不是很多，只有十来个sql使用到了join user。。到时使用代码做join解决。。。但具体还需要分析

## 分库分表user

但也面临join问题 ，和死锁问题，但可以吧死锁概率降低

# 改代码模式

目前已经调整了不少代码，但对死锁解决不能根除