Atitit mysql sql对优化总结

目录

[1. 更新类 1](#_Toc8039)

[1.1. 单条更新类sql 设置事务刷盘模式 1](#_Toc29553)

[1.2. 批量更新 sql类，开启多语句传输multi query模式 1](#_Toc13761)

[1.3. 跳过unique与外键检查 2](#_Toc1461)

[2. 查询优化 4](#_Toc6814)

[2.1. 单条sql查询类，设置索引与查询缓存 25% 4](#_Toc12613)

[2.2. 业务表cache预热 4](#_Toc543)

[2.3. Mybatis 二级缓存集成redis 4](#_Toc21077)

[2.4. 覆盖索引使用 4](#_Toc21972)

[2.5. 慢查询 使用 mysql对prefer schmaer和](#_Toc31137)*[log4jdbc解决](#_Toc31137)* [4](#_Toc31137)

[2.6. 循环大表数据oom问题，使用流式api解决 5](#_Toc9044)

[3. 所有类 5](#_Toc18203)

[3.1. 大批次循环记录处理，使用同机部署与直连模式 sp 5](#_Toc6289)

[3.2. 所有批量sql优化距离 ，本地部署 3倍提升 5](#_Toc20345)

[3.3. 直连模式socket管道 20-50%提升 5](#_Toc17363)

[4. 调整子流程优先级 主流程突出 30% 5](#_Toc2752)

# 更新类

## 单条更新类sql 设置事务刷盘模式

事务隔离级别为rc

## 批量更新 sql类，开启多语句传输multi query模式

可以在jdbc和mysql url模式俩处开启。。。

****Mybatis实现批量操作的三种方式****

* 基于SqlSession的ExecutorType进行批量操作（一次可以操作大量数据，常用）
* 这个貌似开启事务的，有必要测试

创建sqlSession对象 批处理操作：使用SqlSession里 ExecutorType的BATCH方法 \*/ SqlSession sqlSession = this.getSqlSessionFactory().openSession(ExecutorType.BATCH); PersonMapper personMapper = sqlSession.getMapper(PersonMapper.class); //添加10000条记录 for (int i = 0; i <10000 ; i++) { personMapper.addPerson(new User("jerry","bj")); } sqlSession.commit(); sqlSession.close();

## 跳过unique与外键检查

有一种方法可以在插入数据时跳过检查重复项：

如果您对辅助键有UNIQUE约束，则可以通过在导入会话期间暂时关闭唯一性检查来加快表的导入：

SET unique\_checks=0;

... SQL import statements ...

SET unique\_checks=1;

对于大表，这可以节省大量磁盘I / O，因为InnoDB可以使用其插入缓冲区批量写入辅助索引记录。确保数据不包含重复的密钥。

这不是一个选择，因为将CSV拆分为多个表时，检查唯一性和消除重复项是关键。

另一种选择是在加载数据时不检查外键约束

延迟更新索引

InnoDB更改缓冲

内容

1. [更改缓冲区相关状态变量](https://mariadb.com/kb/en/innodb-change-buffering/" \l "change-buffer-related-status-variables" \o "更改缓冲区相关状态变量)
2. [也可以看看](https://mariadb.com/kb/en/innodb-change-buffering/" \l "see-also" \o "也可以看看)

INSERT，UPDATE和DELETE语句执行起来特别繁重，因为每次更改后都需要更新所有索引。因此，这些更改通常会被缓冲。

页面在[缓冲池中](https://mariadb.com/kb/en/innodb-buffer-pool/)被修改，而不是立即在磁盘上被修改。当删除行时，会设置一个标志，因此不会立即在磁盘上删除行。稍后，更改将由InnoDB后台线程写入磁盘（“刷新”）。已在内存中修改但尚未刷新的页面称为脏页面。数据更改的缓冲称为更改缓冲区。

在[MariaDB 5.5](https://mariadb.com/kb/en/what-is-mariadb-55/)之前，只能缓存插入的行，因此该缓冲区称为插入缓冲区。旧名称仍然出现在多个位置，例如，在[SHOW ENGINE INNODB STATUS](https://mariadb.com/kb/en/show-engine-innodb-status/)的输出中。

可以使用以下设置：

* 插入
  + 仅缓冲区插入操作
* 删除
  + 仅缓冲区删除操作
* 变化
  + 缓冲插入和删除操作
* 吹扫
  + 缓冲在后台发生的实际物理删除
* 所有
  + 缓冲区插入，删除和清除。这是[MariaDB 5.5中](https://mariadb.com/kb/en/what-is-mariadb-55/)的默认设置。
* 没有
  + 不要缓冲任何操作。

修改此变量的值仅影响新操作的缓冲。已经缓冲的更改的合并不受影响。

# 查询优化

## 单条sql查询类，设置索引与查询缓存 25%

Mybatis二级缓存 更加精细 ，可以根据id缓存 ，可集成 redis作为mybatis缓存

## 业务表cache预热

针对sql 日志，找到源码部分，然后搜集归纳，cache预热

## Mybatis 二级缓存集成redis

## 覆盖索引使用

## 慢查询 使用 mysql对prefer schmaer和*log4jdbc解决*

*net.sf.log4jdbc.DriverSpy*

12:36:37.141 [main] INFO jdbc.sqlonly - SELECT ID,PLAY\_gROUP\_iD,NUMBER,OPEN\_tIME,DATE,PRE\_oPEN\_cODE,UPDATE\_tIME,COMPANY\_sHORT\_nAME,TOTAL\_bET\_mONEY,TOTAL\_wIN\_mONEY

FROM sSC\_oPEN\_tIME WHERE ( PLAY\_gROUP\_iD = 6 and NUMBER = '202030' ) limit 0,1

frmlogback

12:36:37.149 [main] INFO jdbc.sqltiming - SELECT ID,PLAY\_gROUP\_iD,NUMBER,OPEN\_tIME,DATE,PRE\_oPEN\_cODE,UPDATE\_tIME,COMPANY\_sHORT\_nAME,TOTAL\_bET\_mONEY,TOTAL\_wIN\_mONEY

FROM sSC\_oPEN\_tIME WHERE ( PLAY\_gROUP\_iD = 6 and NUMBER = '202030' ) limit 0,1

{executed in 8 msec}

## 循环大表数据oom问题，使用流式api解决

# 所有类

## 大批次循环记录处理，使用同机部署与直连模式 sp

存储过程游标

## 所有批量sql优化距离 ，本地部署 3倍提升

## 直连模式socket管道 20-50%提升

# 调整子流程优先级 主流程突出 30%