Atitit mysql sql对优化总结

目录

[1.1. 单条更新类sql 设置事务刷盘模式 1](#_Toc17869)

[1.2. 单条sql查询类，设置索引与查询缓存 25% 1](#_Toc32537)

[1.3. 业务表cache预热 1](#_Toc27616)

[1.4. 批量更新 sql类，开启多语句传输multi query模式 1](#_Toc22329)

[1.5. 循环大表数据oom问题，使用流式api解决 2](#_Toc19731)

[1.6. 大批次循环记录处理，使用同机部署与直连模式 sp 2](#_Toc7430)

[1.7. 所有批量sql优化距离 ，本地部署 3倍提升 2](#_Toc19734)

[1.8. 直连模式socket管道 20-50%提升 2](#_Toc32733)

[1.9. 调整子流程优先级 主流程突出 30% 2](#_Toc30360)

## 单条更新类sql 设置事务刷盘模式

事务隔离级别为rc

## 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

## 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();

## 循环大表数据oom问题，使用流式api解决

## 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%