Atitit 物化视图与触发器性能测试方法 attilax总结

[1.1. 触发器主要影响更新性能。。。 1](#_Toc16974)

[1.2. 临时打开关闭触发器，如果db不支持可以更改条件使其不触发 1](#_Toc10609)

[1.3. 打开定时器更新模式测试 1](#_Toc22721)

[1.4. 性能收集 1](#_Toc27226)

[1.5. 性能收集 代码 2](#_Toc11748)

[2. 参考资料 3](#_Toc13687)

## 触发器主要影响更新性能。。。

可以批量并发测试插入更新性能。。。

## 临时打开关闭触发器，如果db不支持可以更改条件使其不触发

## 打开定时器更新模式测试

## 性能收集

不带触发器 50线程并发

---------执行时间：1766ms,avgtime: 656ms,cnt:598,alltime:392320

触发器下 并发线程50人

-执行时间：2491ms,avgtime: 2066ms,cnt:70,alltime:144638

触发器下 30人

---------执行时间：758ms,avgtime: 2016ms,cnt:169,alltime:340734

20人

---------执行时间：741ms,avgtime: 1319ms,cnt:89,alltime:117392

触发器下 并发线程10人

---------执行时间：805ms,avgtime: 736ms,cnt:89,alltime:65578

## 性能收集 代码

要收集本

/oploggerPrj/src/com/attilax/oplog/util/MatViewTest2\_dpt1.java

package com.attilax.oplog.util;

import java.sql.Connection;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Timer;

import java.util.TimerTask;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.atomic.AtomicLong;

import org.apache.ibatis.session.SqlSession;

import com.attilax.oplog.OperLogUtil;

import cn.freeteam.util.MybatisSessionFactory;

public class MatViewTest2\_dpt1 {

public static AtomicLong all\_int = new AtomicLong(0);

public static AtomicLong cnt = new AtomicLong(0);

public static AtomicLong avg = new AtomicLong(0);

synchronized private static void calcNout(long excTime) {

all\_int.addAndGet(excTime);

cnt.incrementAndGet();

avg.set(all\_int.get() / cnt.get());

System.out.println("---------执行时间：" + excTime + "ms,avgtime: " + avg + "ms,cnt:" + cnt + ",alltime:"

+ all\_int);

}

# 参考资料

Atitit 并发测试耗时时间工具

Atitit 并发测试 （结果报告）