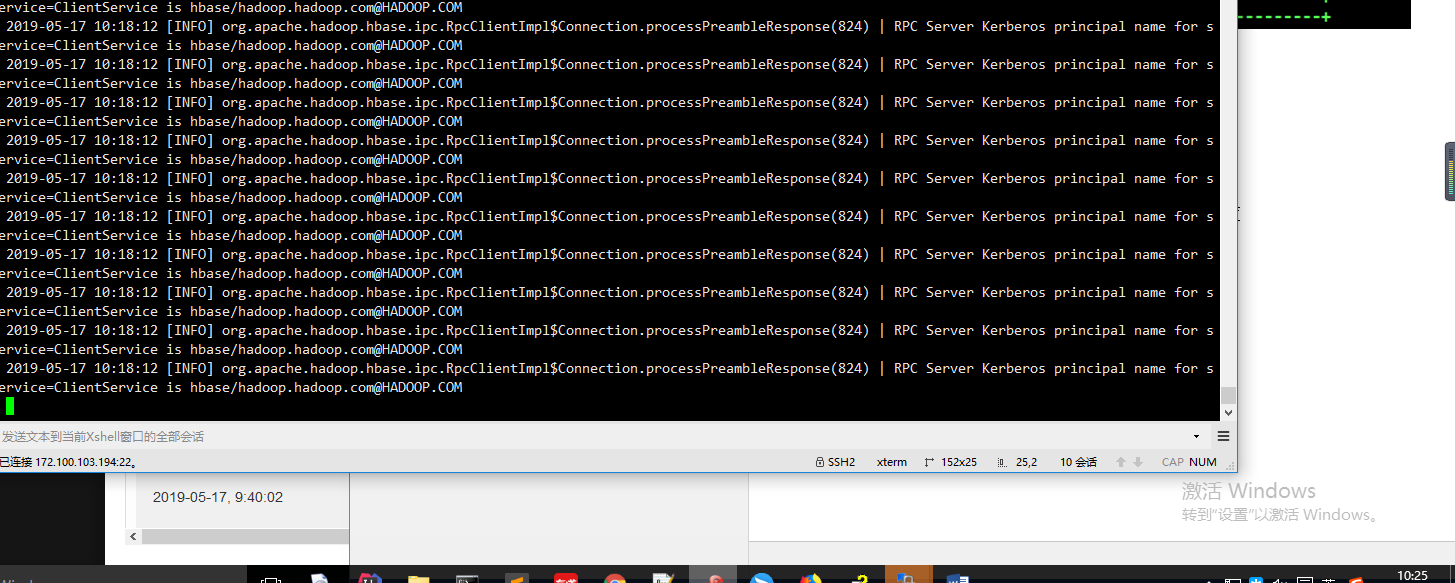
# mtdap-qal同步程序异常

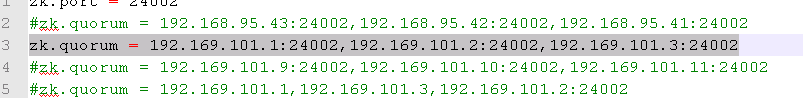
## 1:同步过程中会RPC Server Kerberos

principal name for service=ClientService is hbase/hadoop.hadoop.com@HADOOP.COM



等待大量时间

## 2:权限问题

com.enjoyor.mtdap.hive2phoenix.CongestRoadSynch

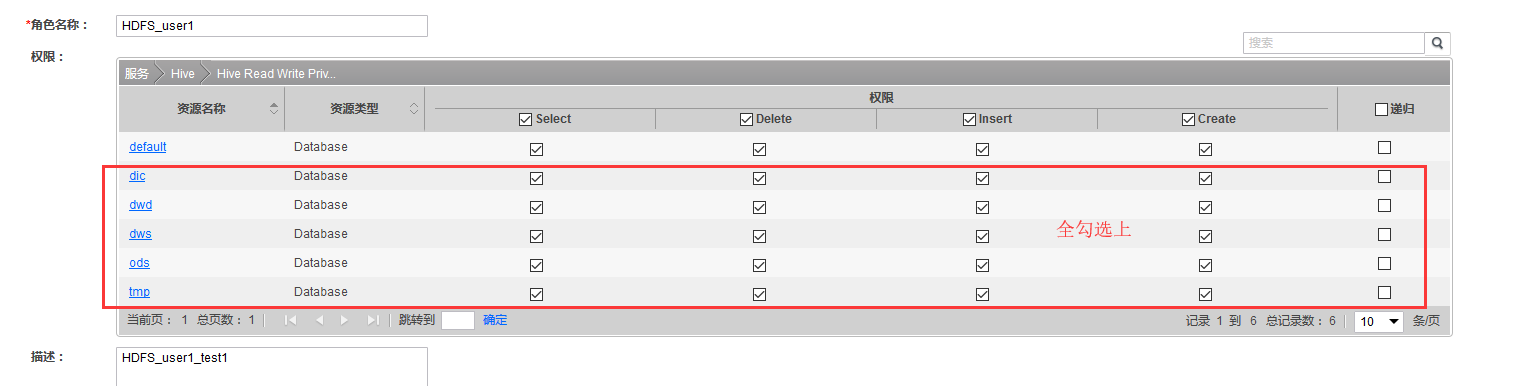
说hdfaadmin用户没有操作以下标的权限

Hdfsadmin用户 有多个角色,但是角色 没有对应的hive新创建表的权限

配置文件不用重新下



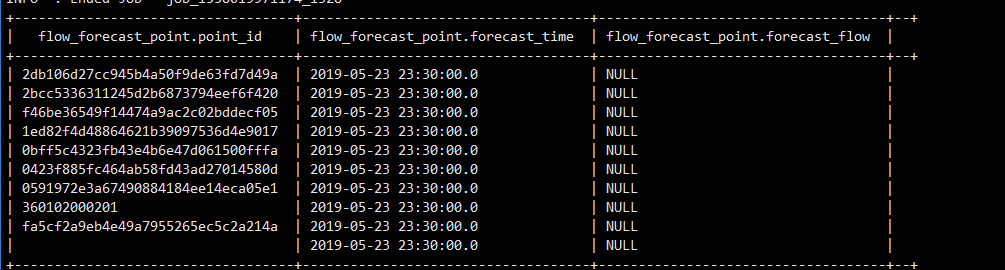




## 3:往redis中写时Caused by: java.lang.ClassNotFoundException: com.huawei.auth.kerberos.ExConfiguration

修改jar包,替换掉华为的jar包

## 4流量预测 为null

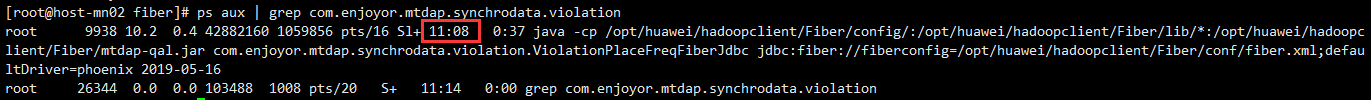


## 5:查看Linux后台执行进程

查看vim sync\_vio.sh脚本执行到哪一步:

查看后台进程,打开sync\_vio.sh脚本中,找到里面程序的关键字 ,比如路径

ps aux | grep com.enjoyor.mtdap.synchrodata.violation 查看



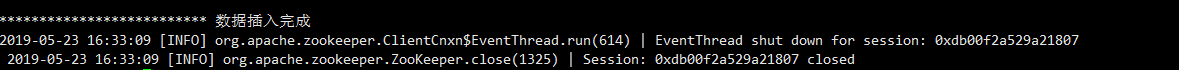
## 6:违法

违法之前的表全部清空,同步5-16号计算到的最新违法数据



之前的数据也是有的,原因是hive中虽然是每天跑几天的数据,但是最后一天是最新的数据,只要同步这一天的就行

## 7:hive->solr数据同步

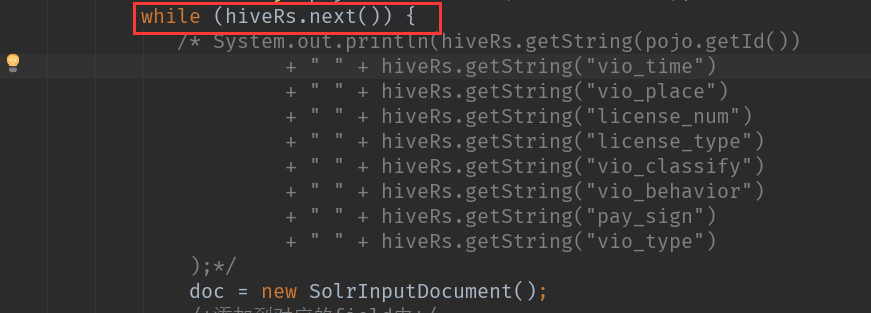


违法数据现在只有4-2日

目标表数据比较少 2019 0 查询 dwd.base\_vio\_violation只有6条 同步失败?

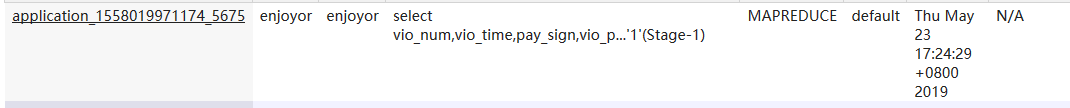
2019 1 168条

发现,没有查到数据 !!



原因;之前是 0 1 选择表在前 日期在后 参数传反了,没有year分区1的,所以空跑

一直卡着:



默认是队列default

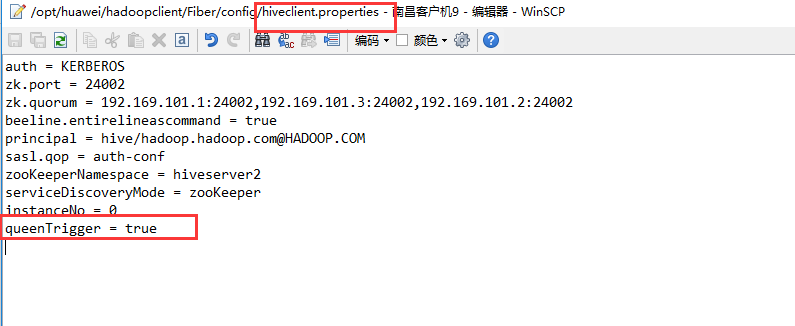
走分区就需要走yarn 全部查询不需要走yarn可以查询

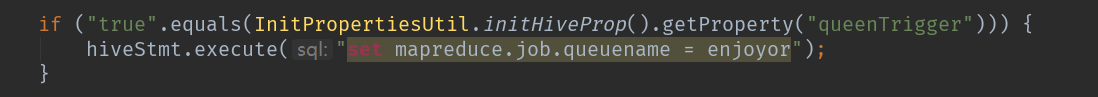
## 8.同步从hive中查询数据需要走队列enjoyor

简单查全部或查分区 后面没有where 不走yarn

其余走yarn

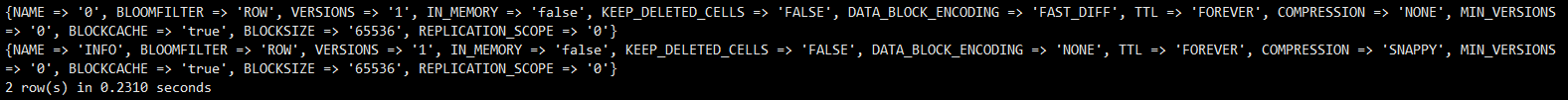
配置文件作为触发器,所有类中都加载上

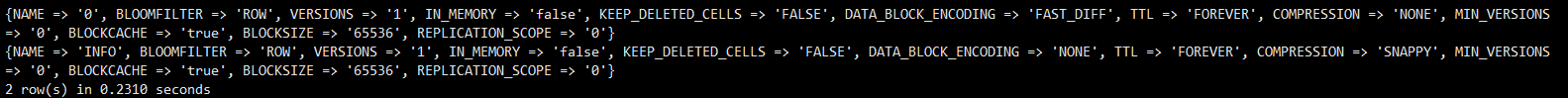




if ("true".equals(InitPropertiesUtil.*initHiveProp*().getProperty("queenTrigger"))) {  
 hivestmt.execute("set mapreduce.job.queuename = enjoyor");  
}

## 9:建表语句





create 'PARAMETER\_CONF',{NAME => 'INFO',COMPRESSION => 'SNAPPY'},{NAME => '0',COMPRESSION => 'SNAPPY'}

去掉后面的:

create 'PARAMETER\_CONF',{NAME => 'INFO',COMPRESSION => 'SNAPPY'}

## 10.Kerberos TGT bug

离线计算程序 在周末两天会算不出来数据

测试;

Acquire TGT from Cache Credentials are no longer valid  
Principal is null null credentials from Ticket Cache  
[Krb5LoginModule] authentication failed   
No password provided

平时:

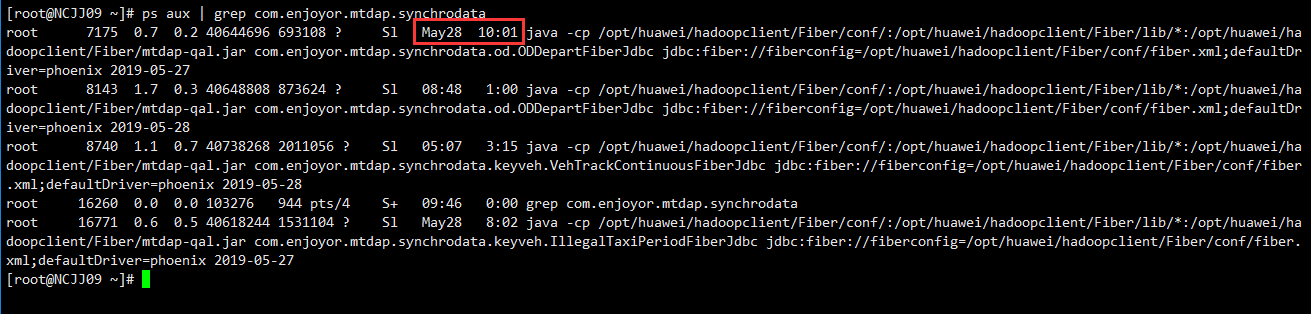
Acquire TGT from Cache Principal is enjoyor@HADOOP.COM  
Commit Succeeded

Kinit –R

KDC can't fulfill requested option while renewing credentials

Kerberos认证超期

## 11:往HBase写入数据越来越慢



查看数据量: 144W

查看告警:



可能的原因:

**1:rowkey设计的问题和块大小设置过小，频繁的合并分拆数据造成的,hbase的使用要尽量设计得分散写、集中读**

2:Phoenix 表未分区的情况下,所有数据会先被写入一个节点,随着数据量的增加,RegionServer对Region拆分,拆分伴随大量的网络IO,对写数据产生极大的影响

每次写入

创建预分区: SALT\_BUCKETS 创建表将表预分割在20个region上

CREATE TABLE SALT\_TEST (a\_key VARCHAR PRIMARY KEY, a\_col VARCHAR) SALT\_BUCKETS = 20;

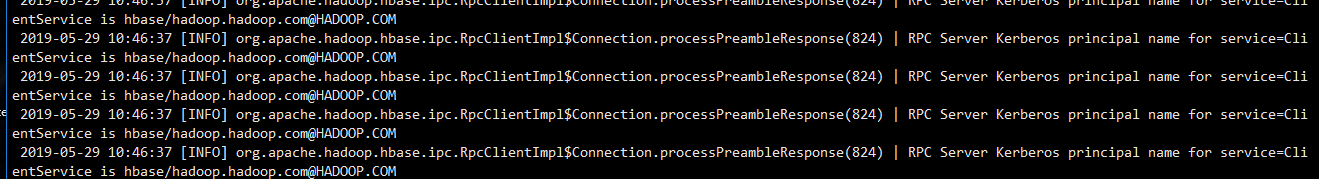
公司表都已将加了,phoenix 预分区已经优化过了 2排除

**测试:**

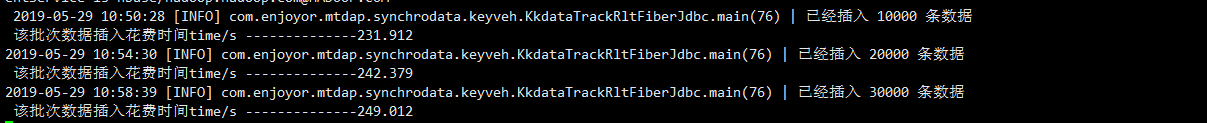
以九江 450w数据的dws.kkdata\_track\_rlt 做测试 ,打印每一次同步时间

同步:卡口关联信息表 1w条数据同步一次

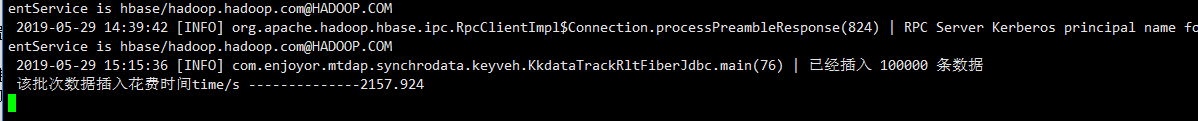
java -cp /opt/huawei/hadoopclient/Fiber/conf/:/opt/huawei/hadoopclient/Fiber/lib/\*:/opt/huawei/hadoopclient/Fiber/mtdap-qal-test.jar com.enjoyor.mtdap.synchrodata.keyveh.KkdataTrackRltFiberJdbc "jdbc:fiber://fiberconfig=/opt/huawei/hadoopclient/Fiber/conf/fiber.xml;defaultDriver=phoenix" 2019-05-27



1w/批 花了231.912秒



10w/批 花费; 预计2157s



查看建表语句:

--卡口轨迹关联表

create table mtdap.kkdata\_track\_rlt

(

record\_id bigint primary key,

track\_id varchar(50)

) default\_column\_family='dcf', salt\_buckets=120, compression='Snappy', immutable\_rows=false, immutable\_rows=false,immutable\_storage\_scheme = one\_cell\_per\_column;

IMMUTABLE\_ROWS=false,不可变索引 即数据不会进行update false表示可变索引,true表示的是不可变索引

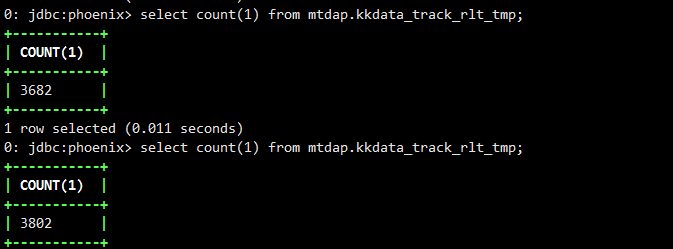
salt\_buckets=120, 设置预分区

重点车辆表有问题? 违法表是没有多个immutable\_rows参数的,

代码同步一个1w条的违法测试下

测试:

改个名字建个表,不建索引表,将 同步immutable\_rows=true去掉, 测试一下1w条的性能,

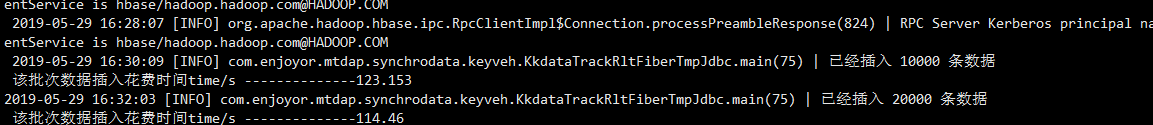


同步是N条N条添加的!,虽然提交时批量1w提交,但是写是分批写入的



不是建表多加一个immutable\_rows参数的问题

换成联合主键同步,查看速度:联合主键速度是索引表的一倍)



先将重点车辆中同步慢(数据量大)的表删掉 改成联合主键

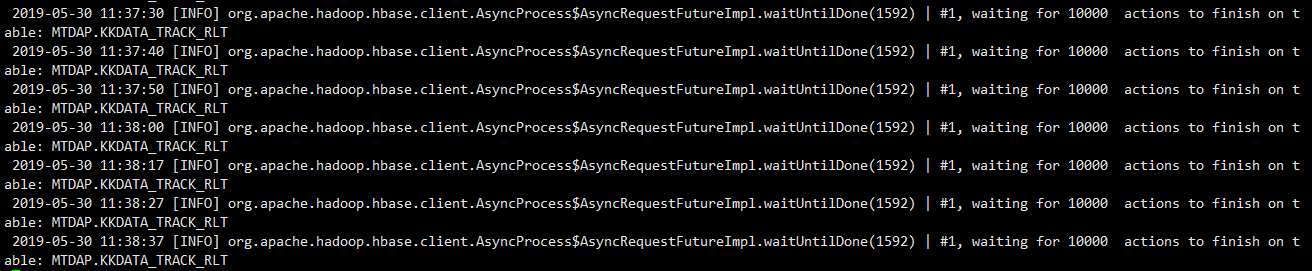
### Jdbc测试

使用jdbc程序重写一个同步的代码,同步1w只需要3-5秒钟

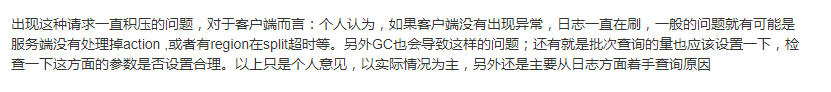
先将慢的使用jdbc的重新实现下

南昌: 27 2867w

同步下:

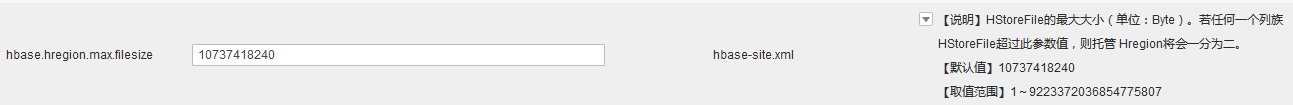




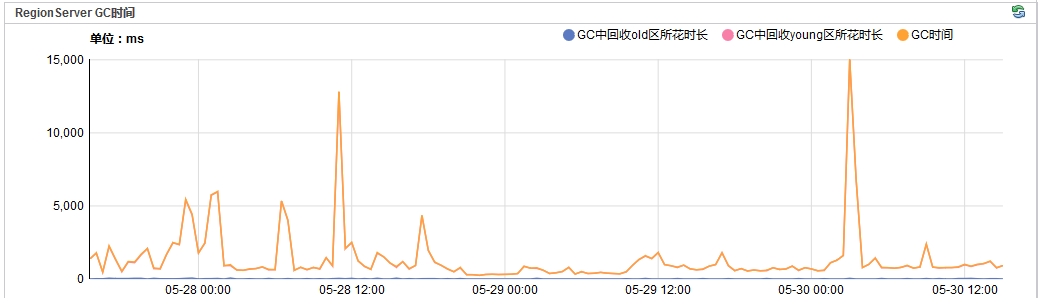


原因:插入的太快 服务端没处理完action ,或者有region在split超时等。另外GC也会导致这样的问题

1:排除region split 10G



2:GC(看人家的100ms)

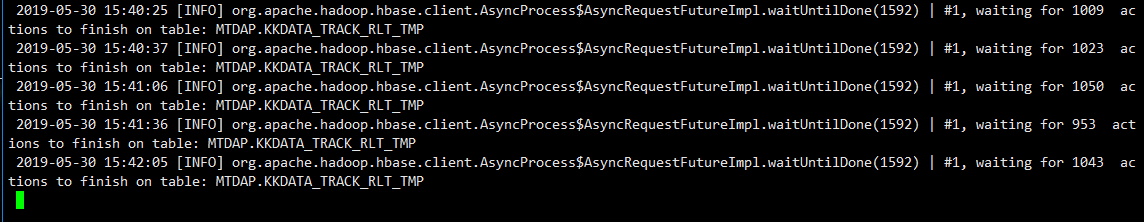


GC平均1s多

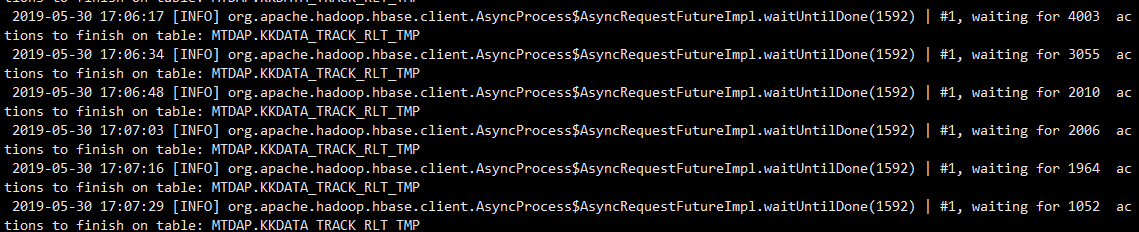
是表里库存多导致的,还是一次性数据拿到太多导致的?

测试表: 2170w/天

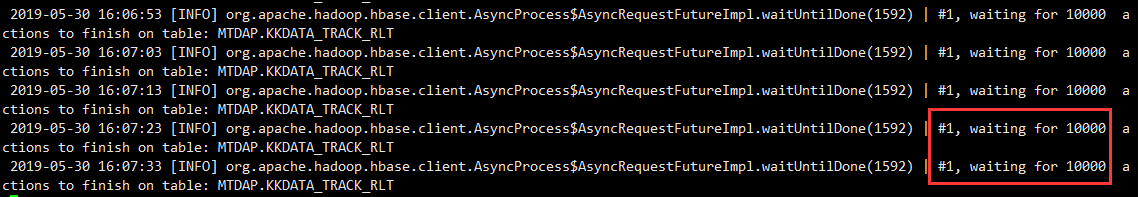
1:建张空表测试(复合主键) 1w提交一次



同步一段时间:等待server服务的时间会增加,然后波动下降到平均值



2:删除原来的表数据(索引表) 1w提交一次

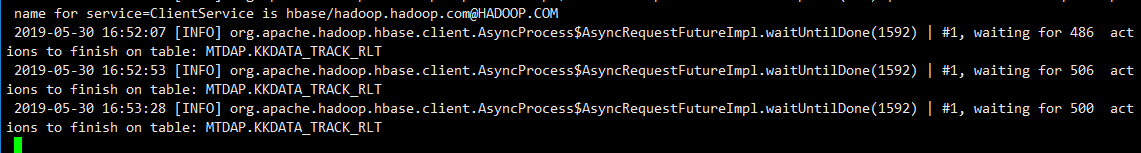


索引表不能要,删除

一次查询2000w数据 复合主键+jdbc 30分钟200w条数据 6-7w/min 1w/10s

60分钟 420w 90分钟/610w 180分/1460w 6h/2500w

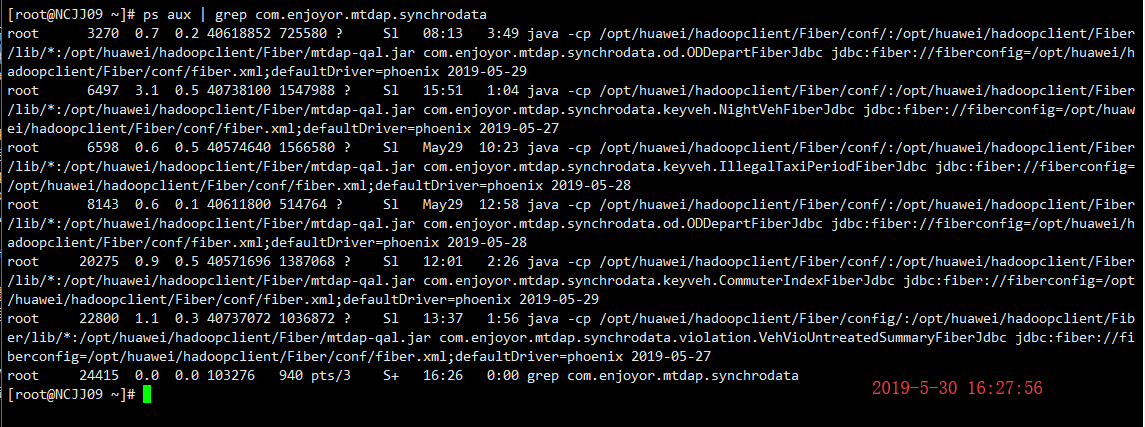
1:批量修改为5000条 索引表 差别不大



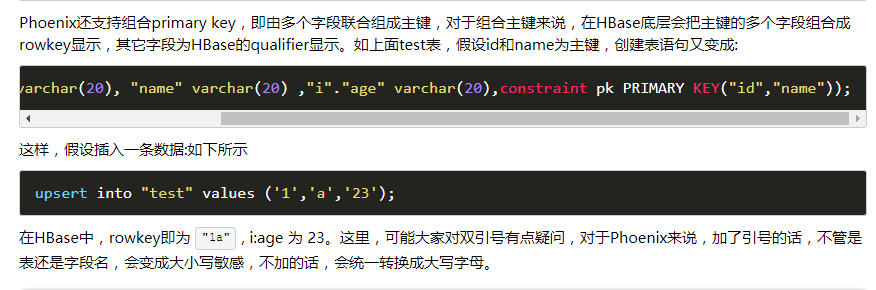
每秒1000-2000/s 4:10分 5:19分 76w ()

2:原有fiber程序测试: 复合主键+fiber 7分钟7k条(废弃fiber)

后台运行程序:



组合主键:把多个字段组成rowkey



## 12:消费者程序严重滞后

Solr 数据抽取 跟踪下(不确定)

车辆轨迹跟车辆点位都滞后 一秒20条大概 ,应该是每条数据太大,poll 2000拿到的数据太大 数据堆积引起的

改成多个程序同时消费一个消费者组,后面改成多线程消费