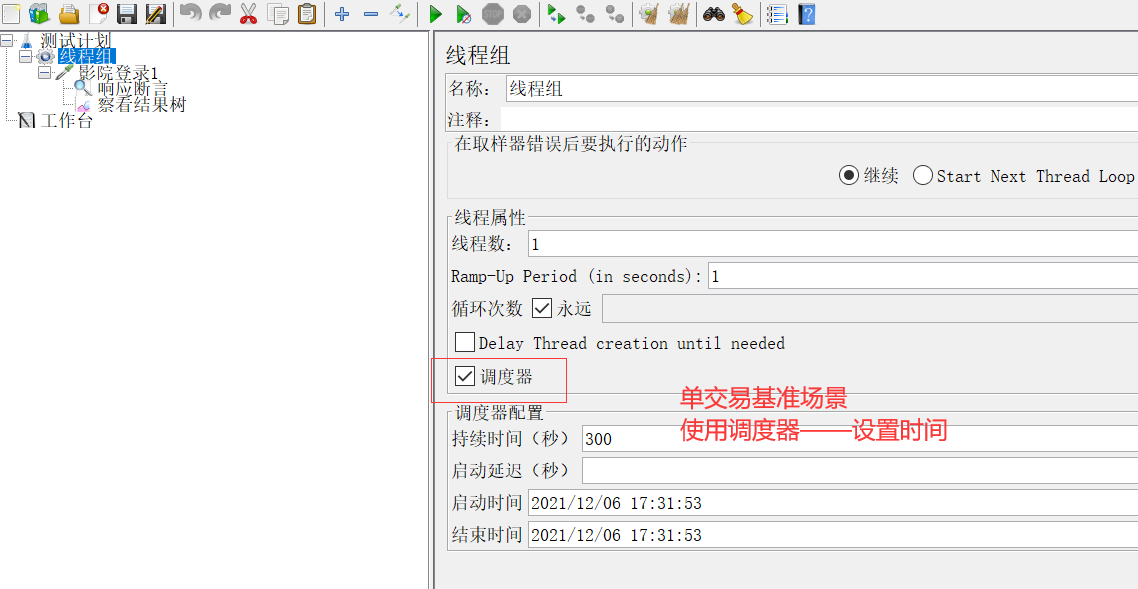
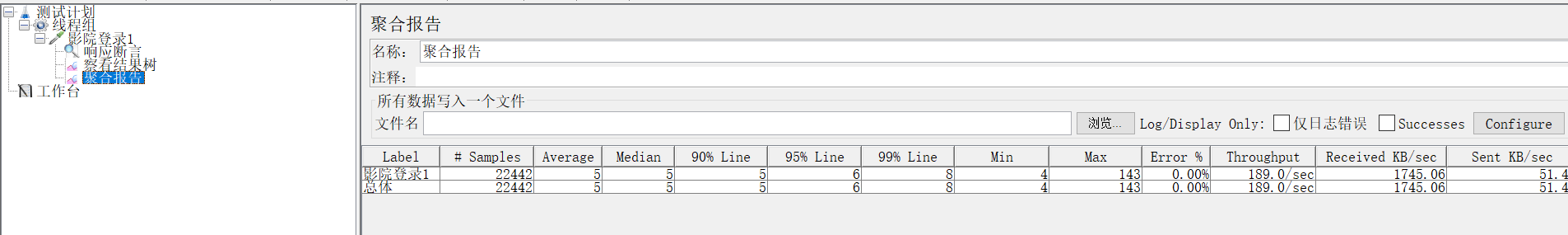
1. 单交易基准场景：1个用户跑一个交易，5-10min



2、收集结果：聚合报告：



Label：请求的名称

Samples：总的请求数

Average：平均响应时间（单位：毫秒）

Media：50%用户的平均响应时间小于该值（单位：毫秒）

90%line：90%用户的平均响应时间小于该值（单位：毫秒）

Min：

Max：

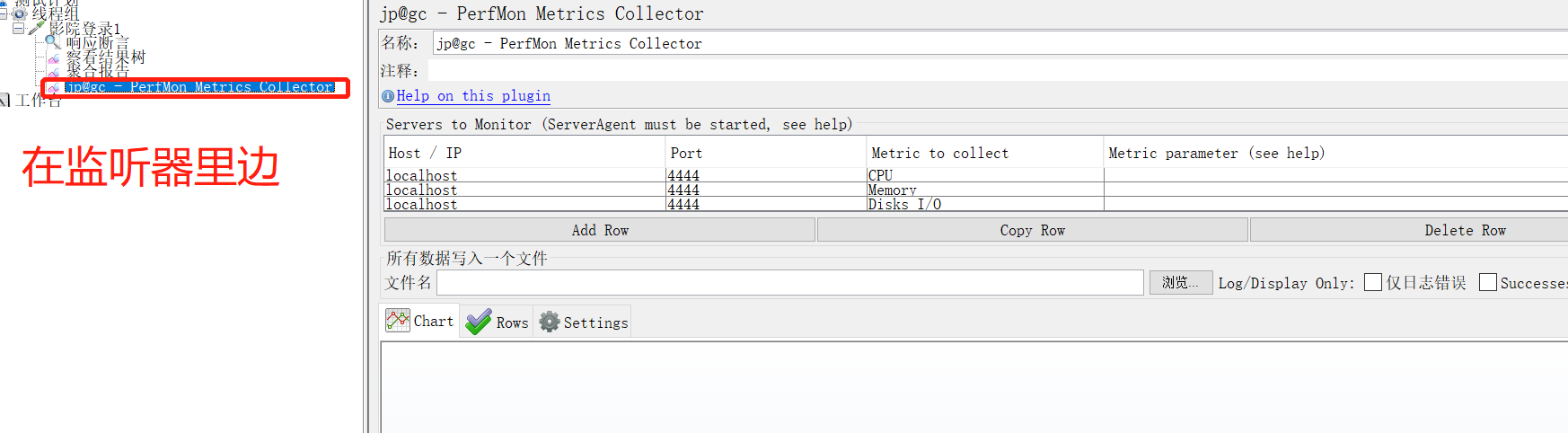
Error：错误率

Throughput：吞吐量（单位：秒）

Received kb/seconds：单位时间内服务端接收到的请求数量（单位：秒）

1. 收集cpu、磁盘、内存perfmon metrics collector

使用前提：必须开启serveragent服务

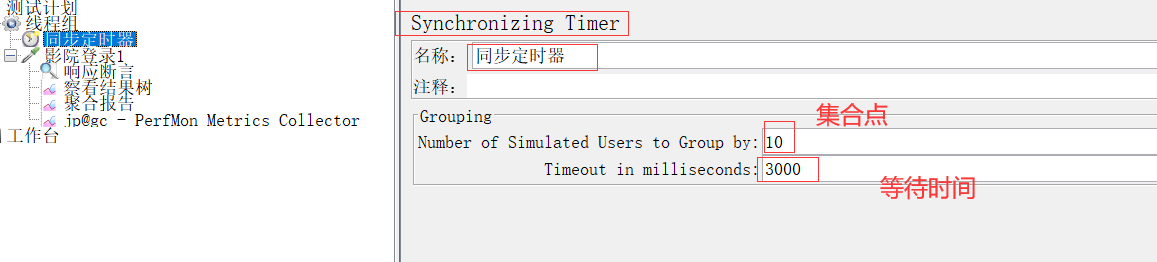


1. 单交易负载场景：

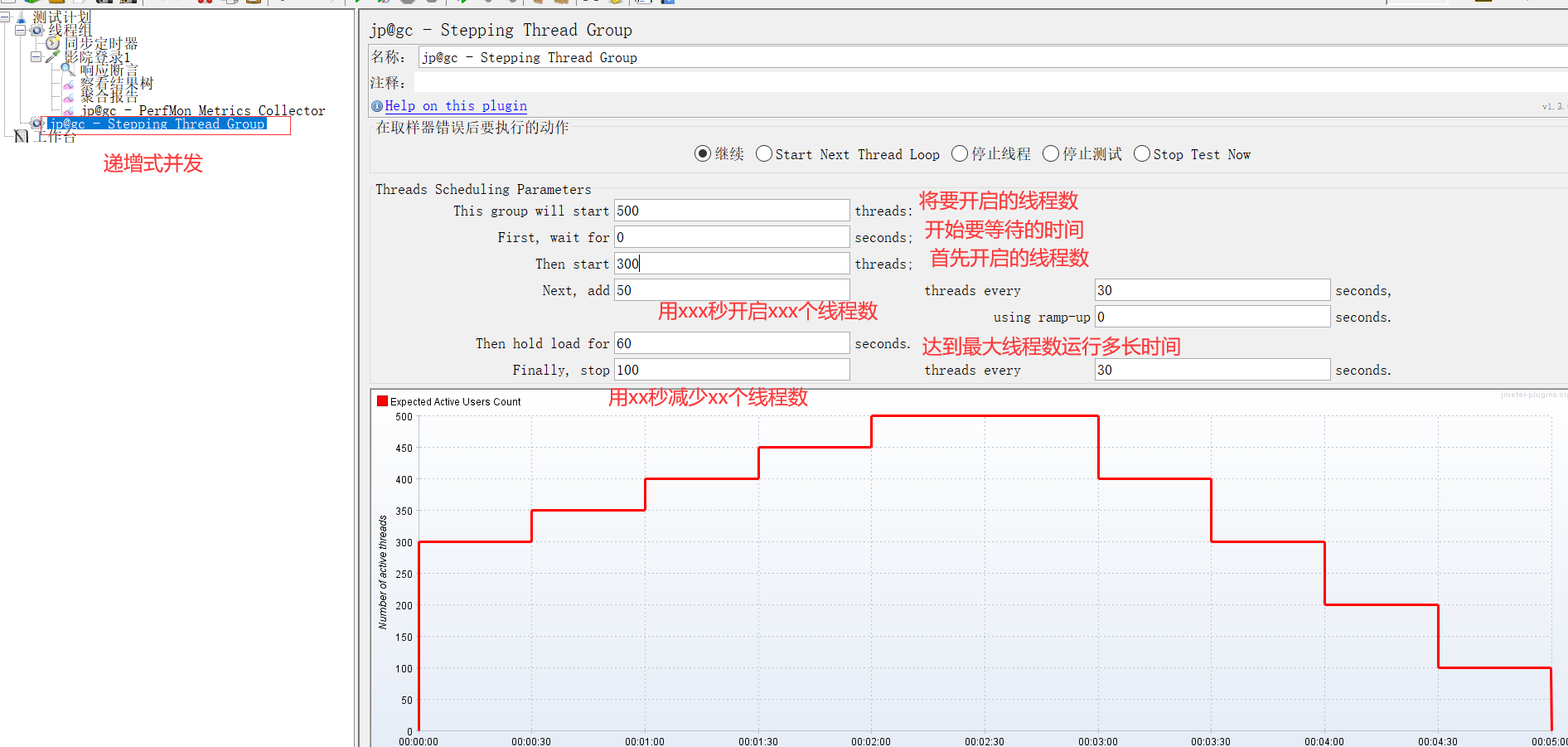
同步并发：同步定时器

固定等待：线程数大于集合点，且等待时间为0

隐式等待：只要到了等待时间就会执行



递增式并发：setping thread group

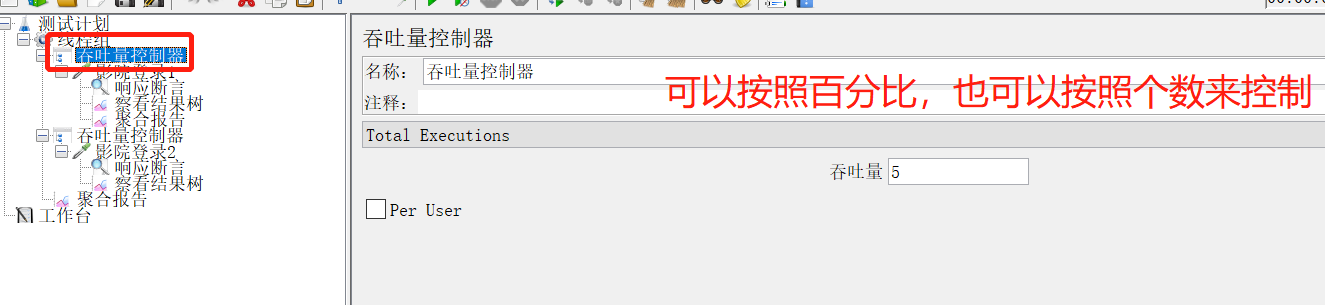


1. 混合测试场景：多个用户跑多个交易（压测比）

压测比的确定：

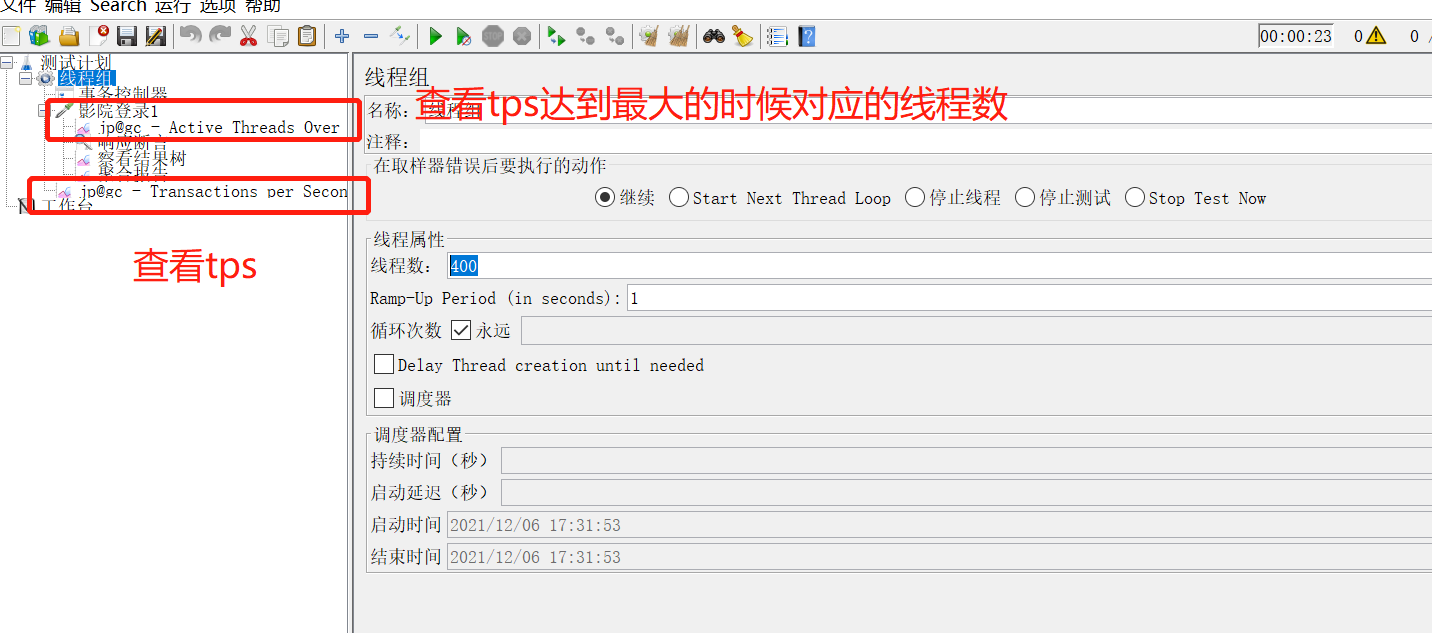
1. 已上线的系统：直接去找运维或者大数据分析师
2. 未上线的系统：
   1. 系统的核心功能
   2. 用户使用最频繁的
   3. 发行一个公测版本
   4. 产品或者需求调研人员去调研

控制压测比：吞吐量控制器

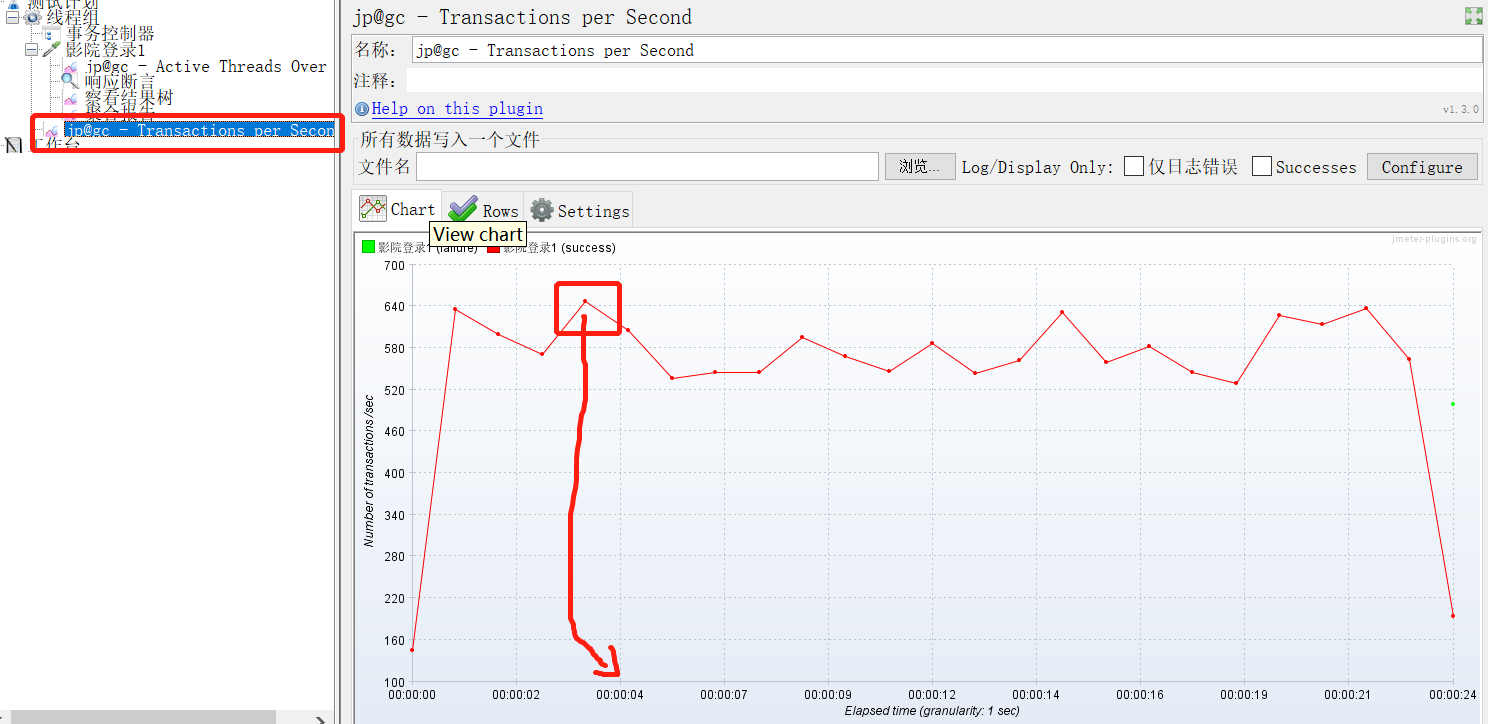


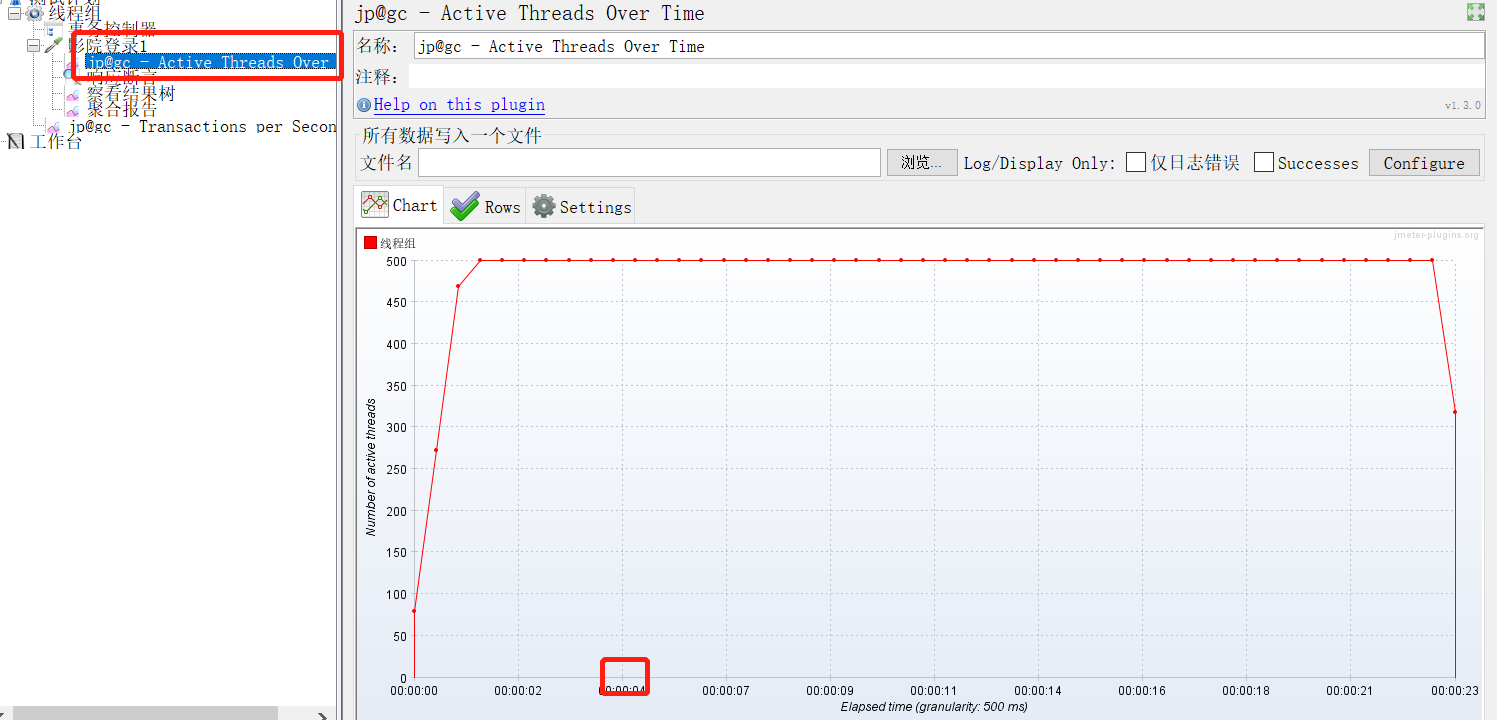
1. 稳定性测试场景

混合测试场景最大压力的80%（tps达到最大的时候对应的线程数）执行4h或者12h



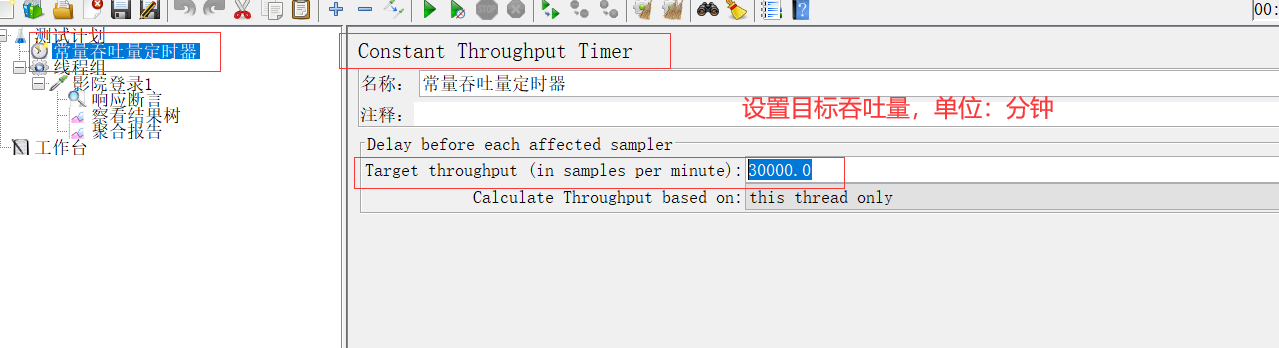
Tps达到最大的时间点

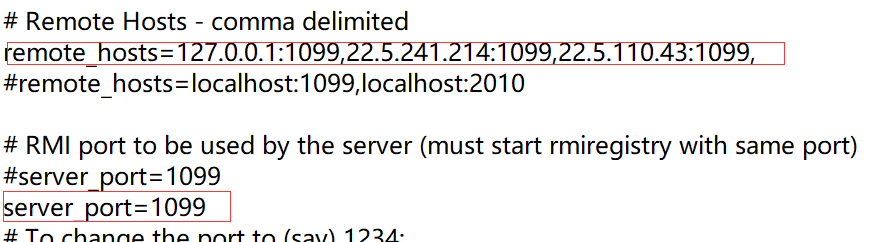


查看tps最大时候对应的线程数  


1. 基于目标的场景（常量吞吐量定时器）

提出目标：查看吞吐量达到500/seconds的时候，平均响应时间能不能小于2s



1. 分布式负载测试
2. 调度机：
3. 将每台执行机的ip地址和端口号添加到jmeter/bin目录下的jmeter.properties文件里边remote\_hosts
4. 

3）开启服务：jmeter-server.bat服务

1. 执行机：
2. 只需要将自己的ip地址和端口号告诉调度机
3. 开启服务：jmeter-server.bat服务

注意事项：

1. 调度机和执行机使用的jmeter版本、jdk版本保持一致
2. 调度机中的脚本必须和执行机的脚本存放在同一个路径下

查看端口号：netstat -ano|findstr 要查的端口号

Netstat -ano|grep 要查的端口号