## JMeter入门

1.官网：<http://jmeter.apache.org/>

2.下载：<http://jmeter.apache.org/download_jmeter.cgi/>

3.用户手册：<http://jmeter.apache.org/usermanual/index.html>

### 一、基本概念汇总

并发在多少的时候网站的qps（每秒钟request/事务 数量）是多少

双十一每秒钟完成20万笔订单这叫tps（系统吞吐量）

QPS(QPS每秒查询率(Query Per Second))

概念: 每秒查询率QPS是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准，在因特网上，作为域名系统服务器的机器的性能经常用每秒查询率来衡量。对应fetches/sec，即每秒的响应请求数，也即是最大吞吐能力。 （看来是类似于TPS，只是应用于特定场景的吞吐量）

吞吐量是指系统在单位时间内处理请求的数量。对于无并发的应用系统而言，吞吐量与响应时间成严格的反比关系，实际上此时吞吐量就是响应时间的倒数。前面已经说过，对于单用户的系统，响应时间（或者系统响应时间和应用延迟时间）可以很好地度量系统的性能，但对于并发系统，通常需要用吞吐量作为性能指标。

原理：每天80%的访问集中在20%的时间里，这20%时间叫做峰值时间。

公式：( 总PV数 \* 80% ) / ( 每天秒数 \* 20% ) = 峰值时间每秒请求数(QPS) 。

机器：峰值时间每秒QPS / 单台机器的QPS = 需要的机器 。

每天300w PV 的在单台机器上，这台机器需要多少QPS？

( 3000000 \* 0.8 ) / (86400 \* 0.2 ) = 139 (QPS)。

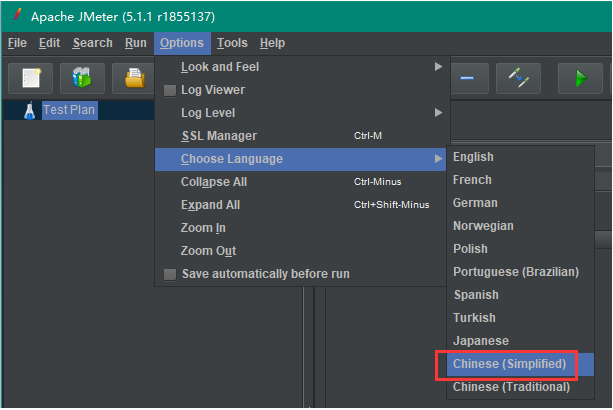
一般需要达到139QPS，因为是峰值。

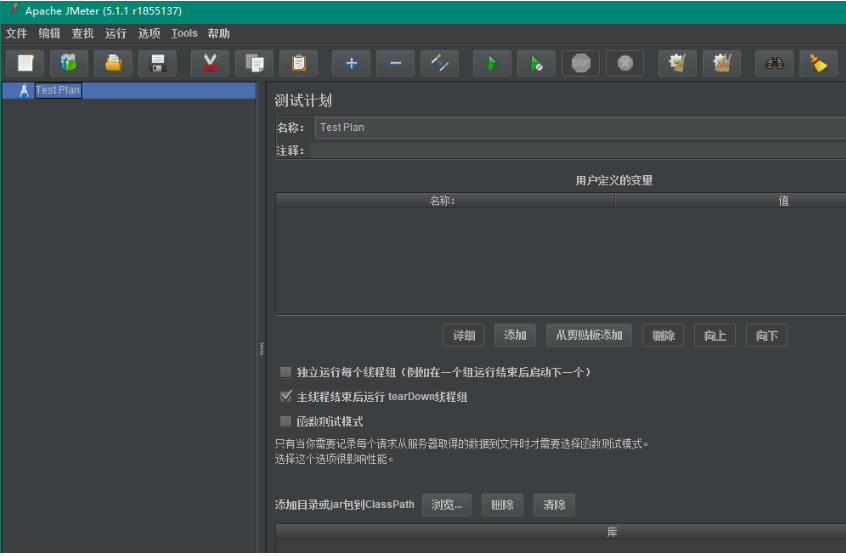
TPS(系统吞吐量)

概念: 对于一个多用户的系统，如果只有一个用户使用时系统的平均响应时间是t，当有你n个用户使用时，每个用户看到的响应时间通常并不是n×t，而往往比n×t小很多（当然，在某些特殊情况下也可能比n×t大，甚至大很多）。这是因为处理每个请求需要用到很多资源，由于每个请求的处理过程中有许多不走难以并发执行，这导致在具体的一个时间点，所占资源往往并不多。也就是说在处理单个请求时，在每个时间点都可能有许多资源被闲置，当处理多个请求时，如果资源配置合理，每个用户看到的平均响应时间并不随用户数的增加而线性增加。实际上，不同系统的平均响应时间随用户数增加而增长的速度也不大相同，这也是采用吞吐量来度量并发系统的性能的主要原因。一般而言，吞吐量是一个比较通用的指标，两个具有不同用户数和用户使用模式的系统，如果其最大吞吐量基本一致，则可以判断两个系统的处理能力基本一致。

### 二、基本设置

1.jmeter设置中文



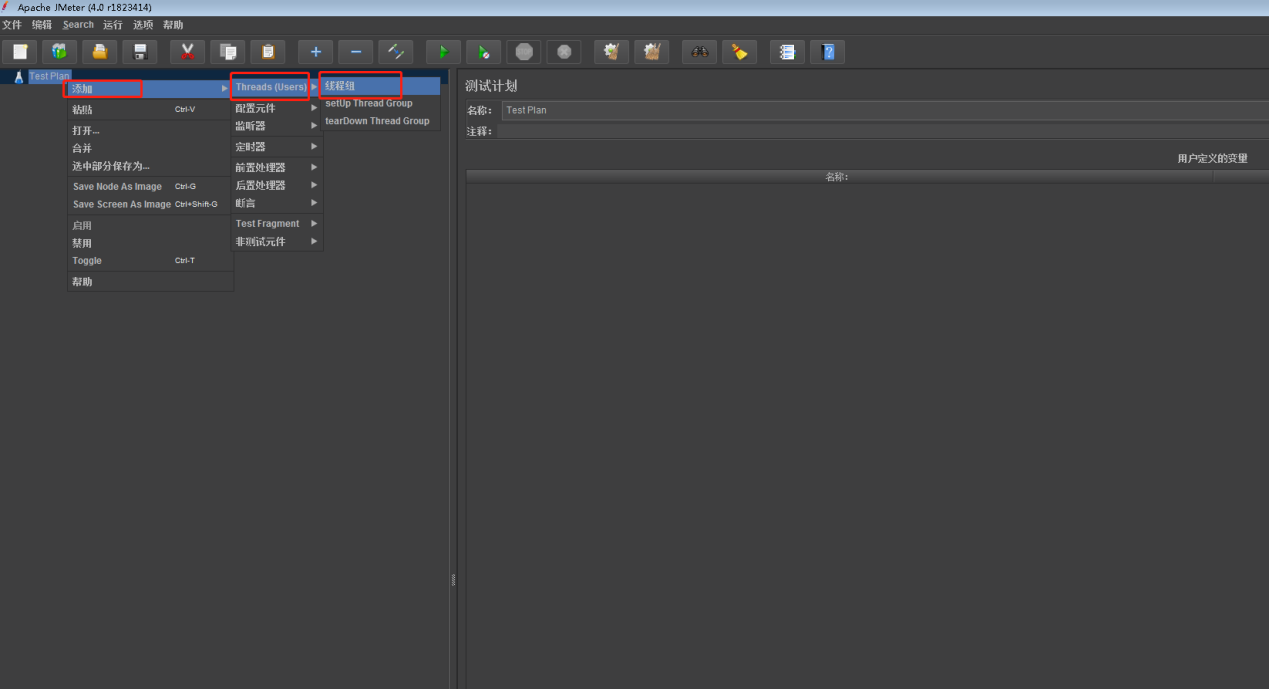


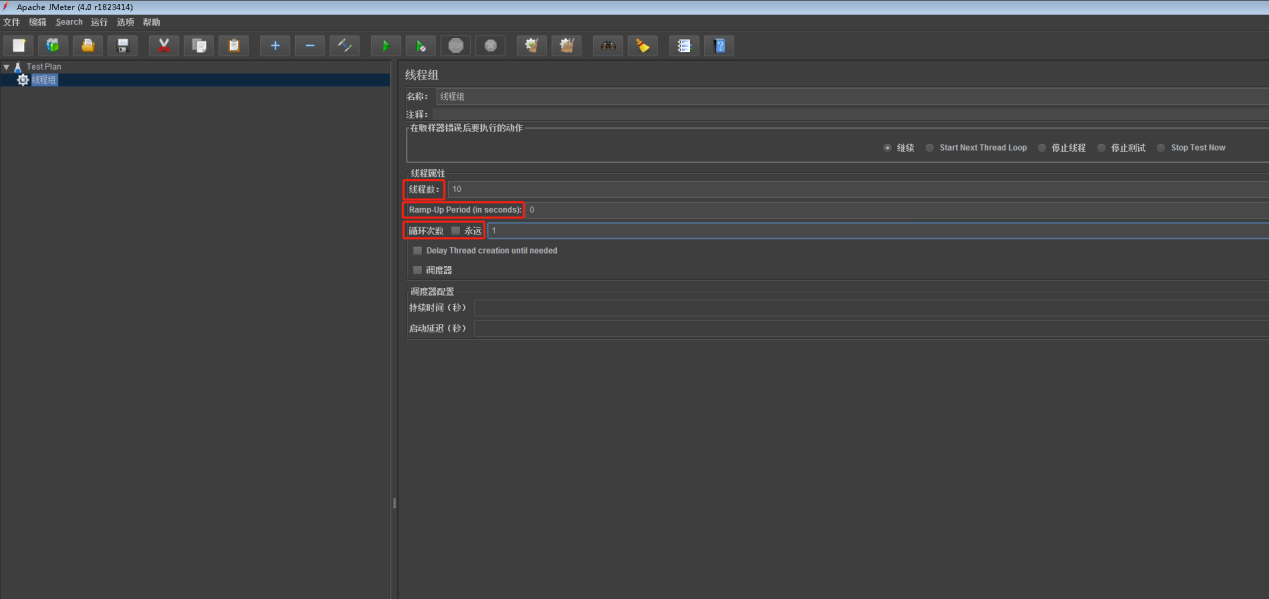
### 三、操作以及测试例子

创建jmeter测试用例测试接口操作如下：

#### 步骤一：创建一个线程组（即线程数量的创建）

打开jmeter，选择测试计划->右键添加->Threads(Users)->线程组





线程属性解释如下：

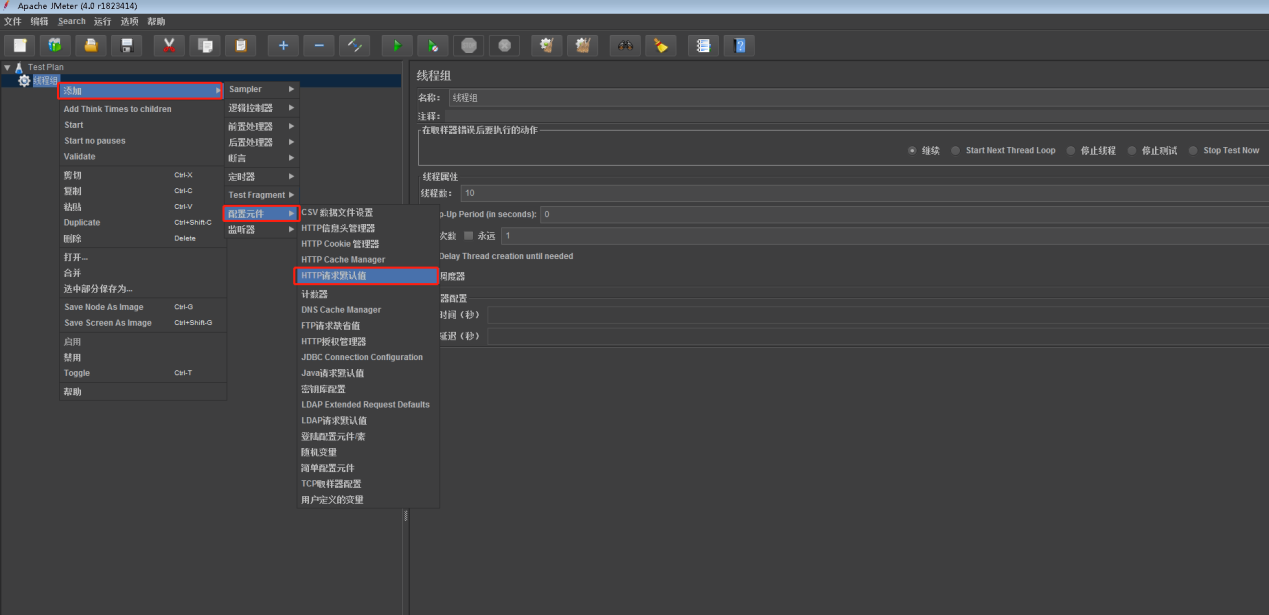
线程数：线程数即我们的并发数，赋值10表示现在使用10个并发操作。

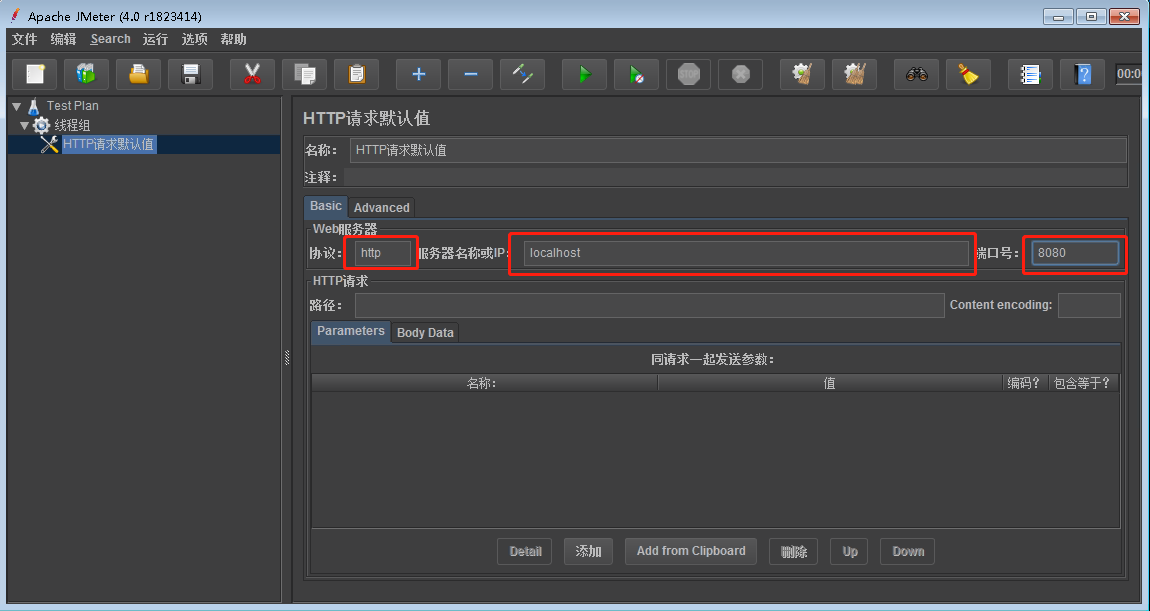
Ramp-Up Period（in seconds）：表示线程数用多长时间启动起来，如果图中赋值是1表示1秒钟用户把这10个线程数先后启动起来，如果是0表示这10个线程一快启动起来。

循环次数[]永远：表示我们的线程访问这个测试接口的时候一共循环访问多少次，如果是1表示我们的10个线程数，每一个线程都对测试接口访问1次，如果是10表示我们的10个线程中的每一个线程对指定测试接口循环访问10次，对接口的总共访问次数是100次。

#### 步骤二：创建HTTP请求默认值

选择步骤一新建的线程组->右键->添加->配置元件->HTTP请求默认值



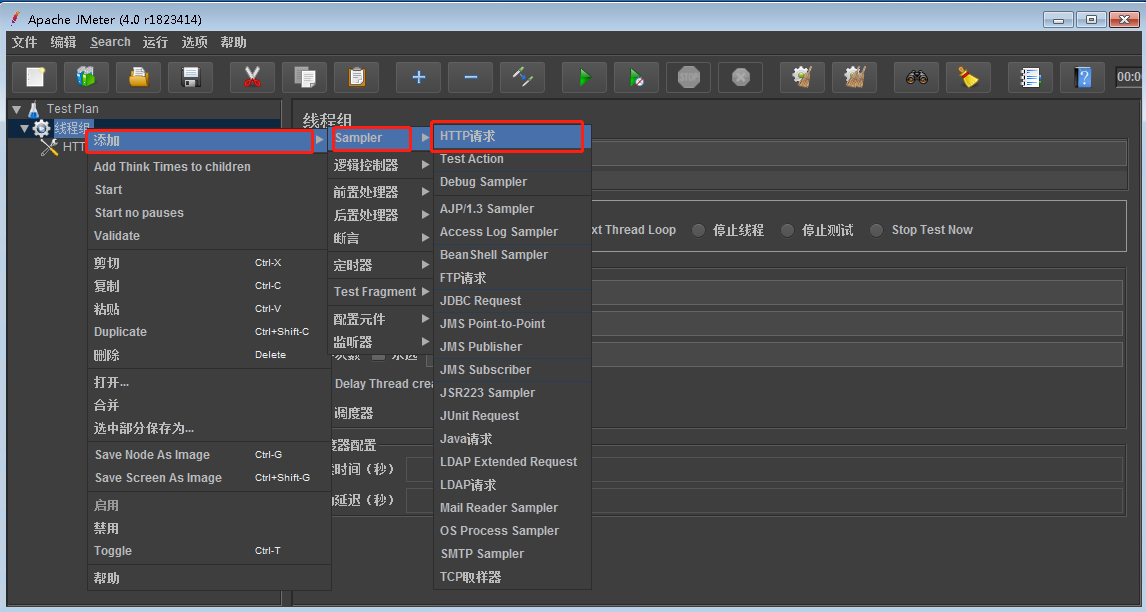


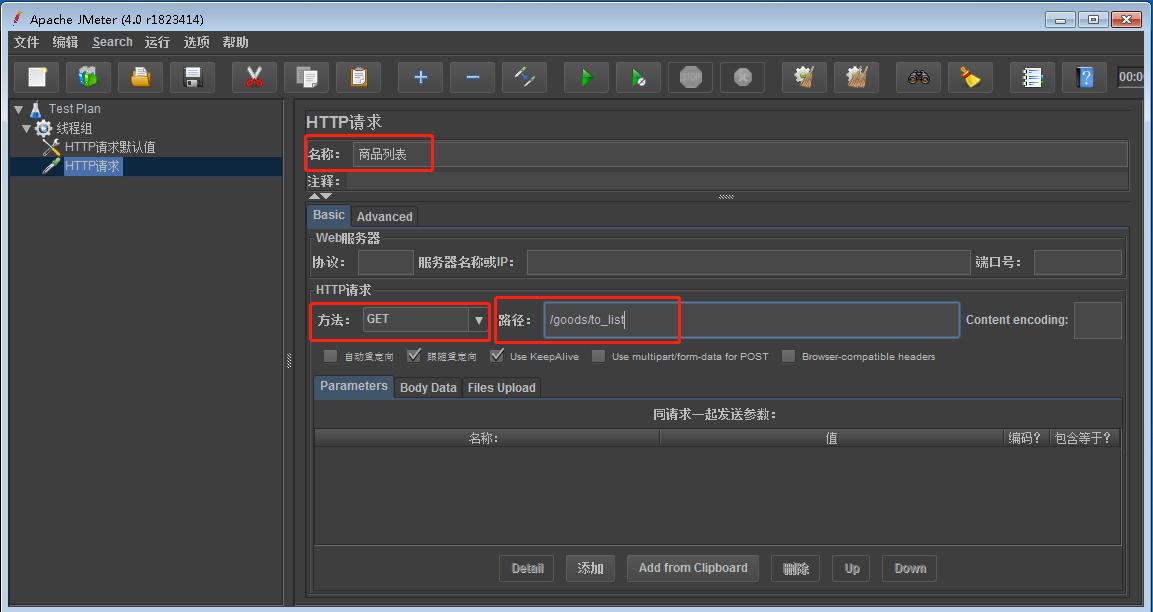
作用以及好处：在HTTP请求默认值中配置了协议、服务器名称或IP和端口号以后其他的请求中就不需要再对协议、服务器名称或IP和端口号进行配置

#### 步骤三：创建一个HTTP请求，访问商品列表接口（Sampler）

选择步骤一新建的线程组->右键->添加->Sampler->HTTP请求

访问商品列表接口





接下来填写名称、方法以及路径,因为该接口不需要请求参数因此这里不需要填写请求参数，带有请求参数的压测后续介绍，注意，这里因为步骤二配置了HTTP请求默认值所以HTTP请求里面的协议、服务器名称或IP以及端口号不需要再填写。

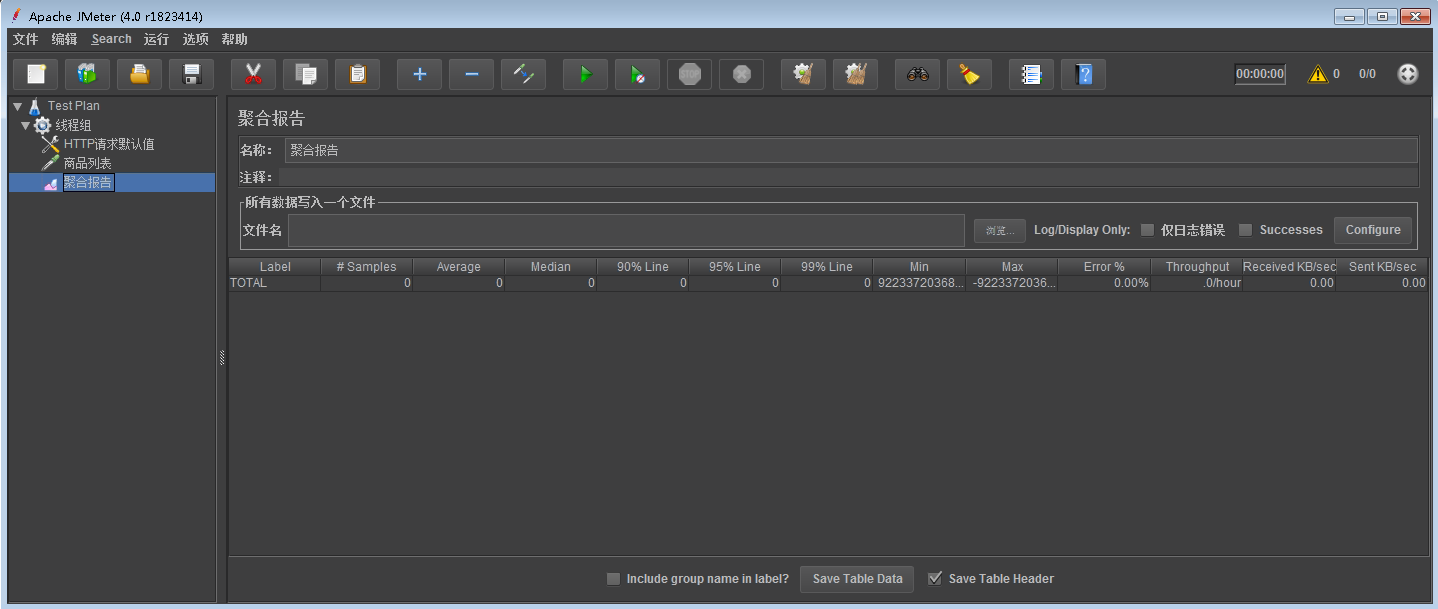
#### 步骤四：创建监听器，查看对指定接口压测后的输出结果

选择步骤一新建的线程组->右键->添加->监听器

##### （1）聚合报告（属于概括说明报告）

选择步骤一新建的线程组->右键->添加->监听器->聚合报告





作用：把我们对指定接口压测的情况以聚合报告的形式展示出来，聚合报告是一个很概括的说明，如果需要查看详细的说明可以通过生成图形结果、结果树、用表格察看结果报告等方式查看。

页面各个字段代表的含义解释如下：

Label: 每个 JMeter 的 element（例如 HTTP Request）都有一个 Name 属性，这里显示的就是 Name 属性的值

#Samples: 表示这次测试中一共发出了多少个请求，如果模拟10个用户，每个用户迭代10次，那么这里显示100；【我的是用户有100，只迭代一次，因此也是100】

Average: 平均响应时间——默认情况下是单个 Request 的平均响应时间，当使用了 Transaction Controller 时，也可以以Transaction 为单位显示平均响应时间

Median: 中位数，也就是 50％ 用户的响应时间

90%Line: 90％用户的响应时间

95%Line: 95％用户的响应时间

99%Line: 99%用户的响应时间

Min: 最小响应时间

Max: 最大响应时间

Error%: 本次测试中出现的错误率，即错误的请求的数量/请求的总数

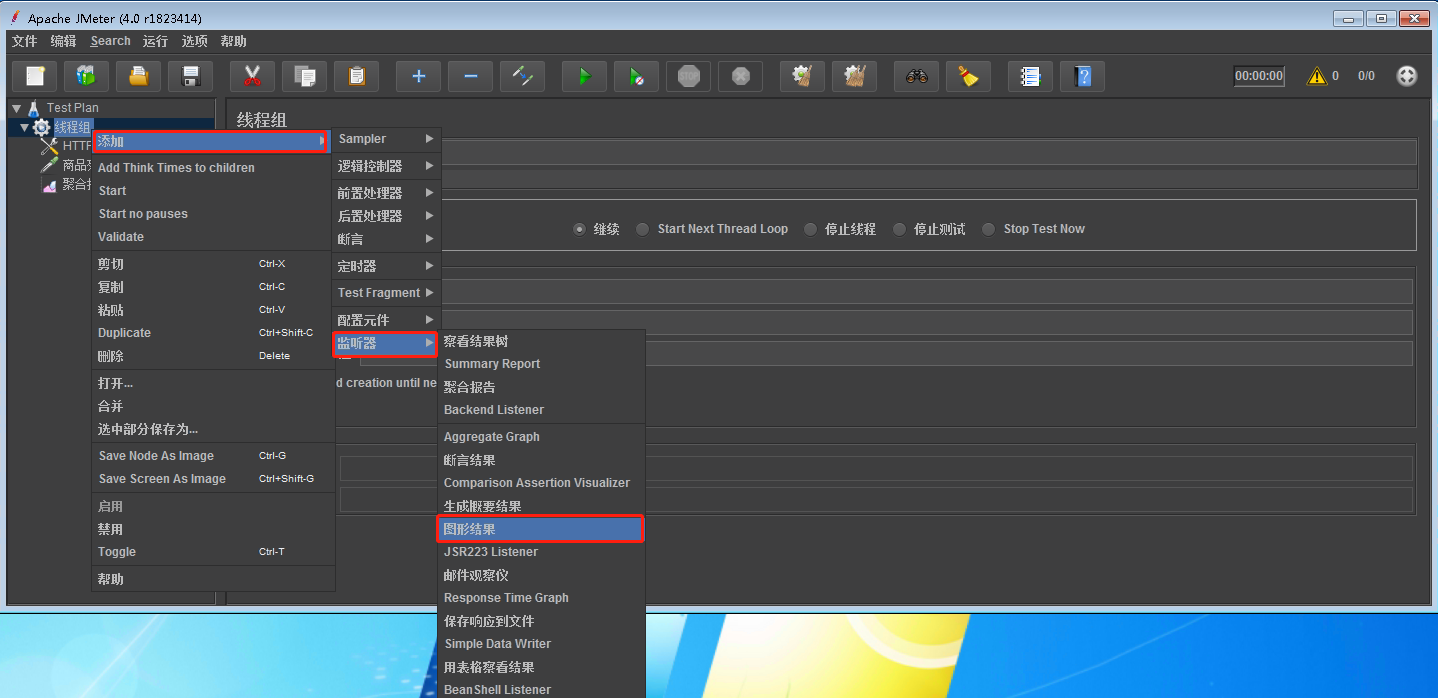
Throughput: 吞吐量(即TPS)——默认情况下表示每秒完成的请求数（Request per Second），当使用了 Transaction(事务)

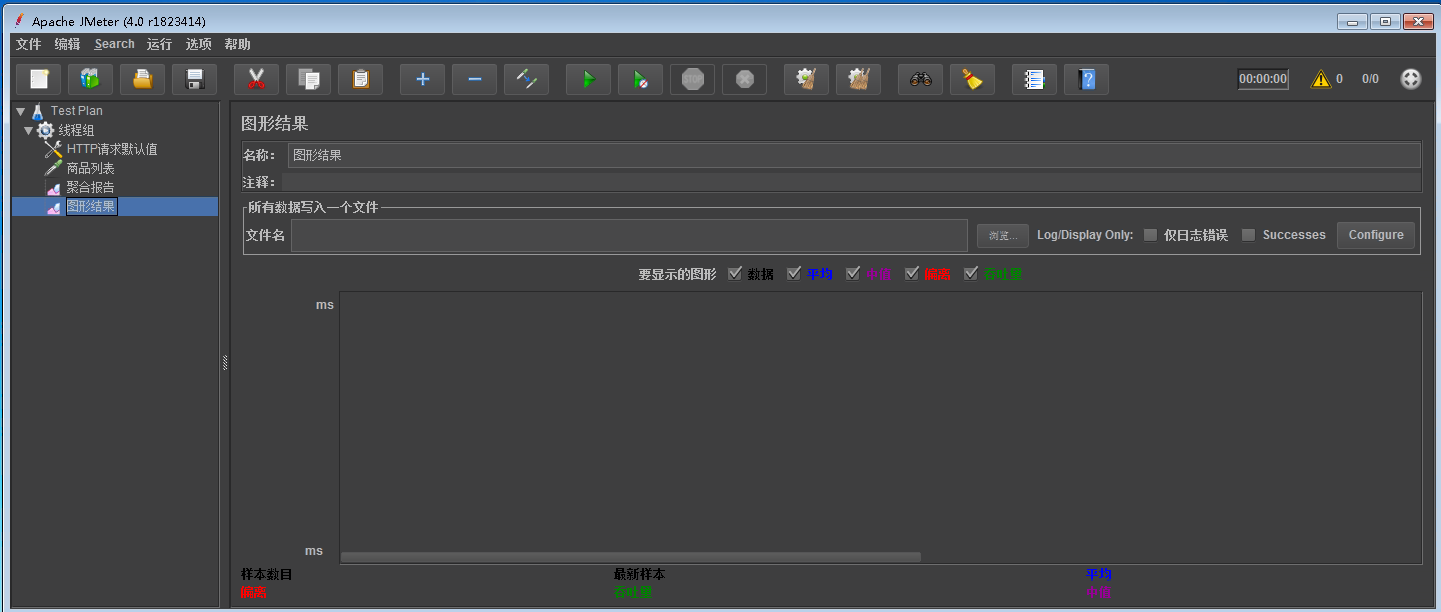
Received KB/sec: 每秒从服务器端接收到的数据量

Sent KB/sec: 每秒从客户端发送的请求的数量

##### （2）图形结果（属于详细说明报告）

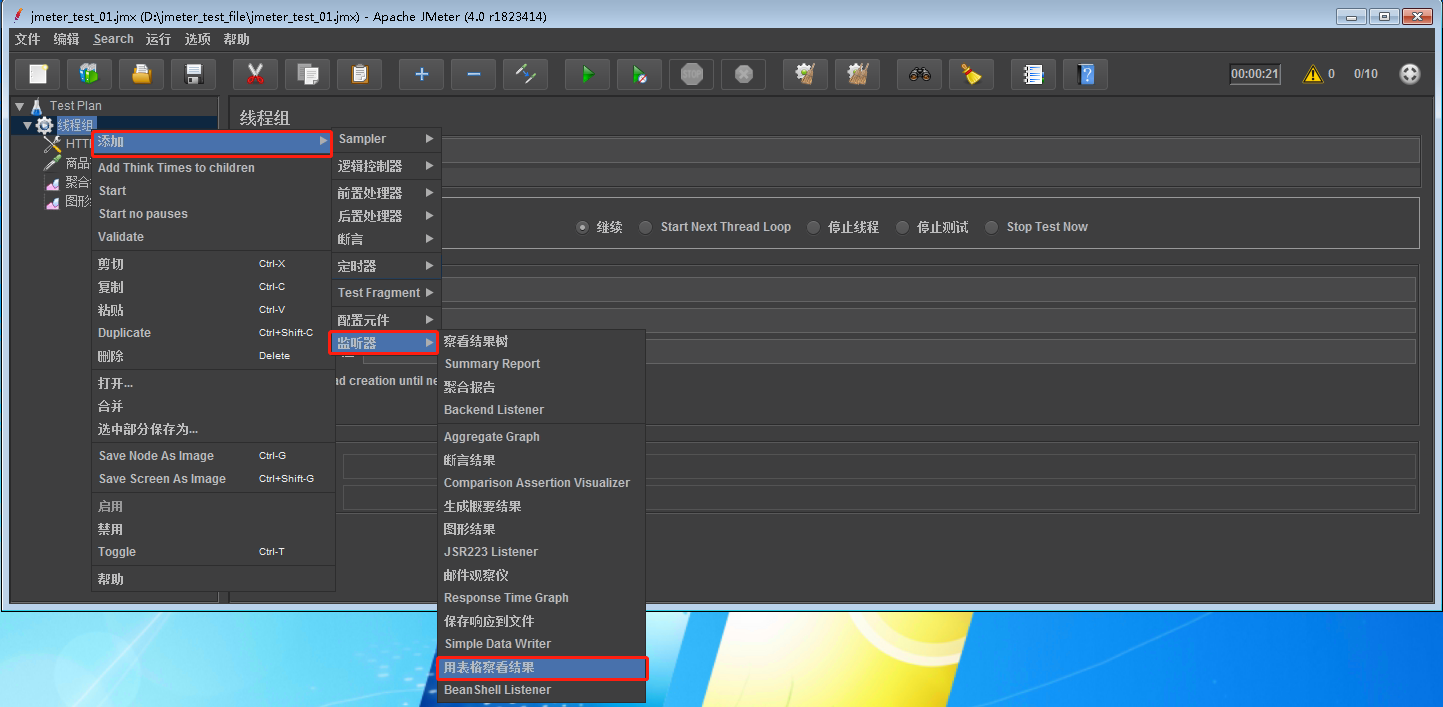
选择步骤一新建的线程组->右键->添加->监听器->图形结果

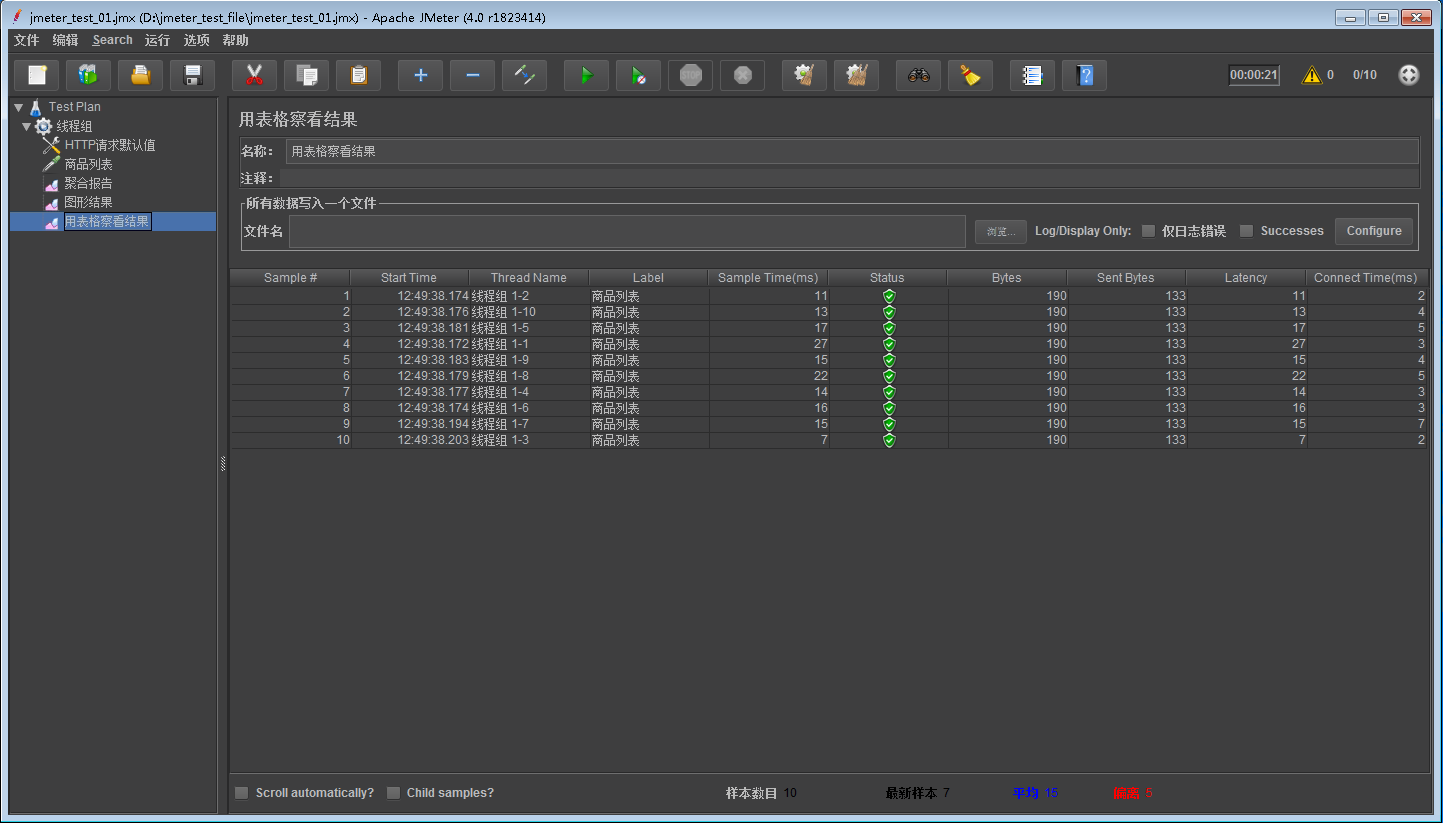




##### （3）用表格察看结果（属于详细说明报告）

选择步骤一新建的线程组->右键->添加->监听器->用表格察看结果





页面各个字段代表的含义解释如下：

Sample#: 每个请求的序号

Start Time: 每个请求开始时间（时：分：秒.毫秒）

Thread Name: 每个线程的名称（线程序号-第N次循环次数）

Label: 每个请求的自定义名称（无修改时默认显示请求类型，如Http，FTP等请求）

Sample Time(ms): 每个请求的响应时间（单位：毫秒）

Status: 请求状态，如果为勾则表示成功，如果为叉表示失败

Bytes: 响应的字节数，请求的字节数

Sent Bytes: 发送的字节数

Latebcy: 延迟的时间，等待时长（单位：毫秒）

Connect Time(ms): 连接服务器的时间（单位：毫秒）

样本数目：所有请求个数，样本数目 = 线程数（请求用户数）\* 请求次数 。（单位：个）

平均：所有请求的平均响应时间。（单位：毫秒）

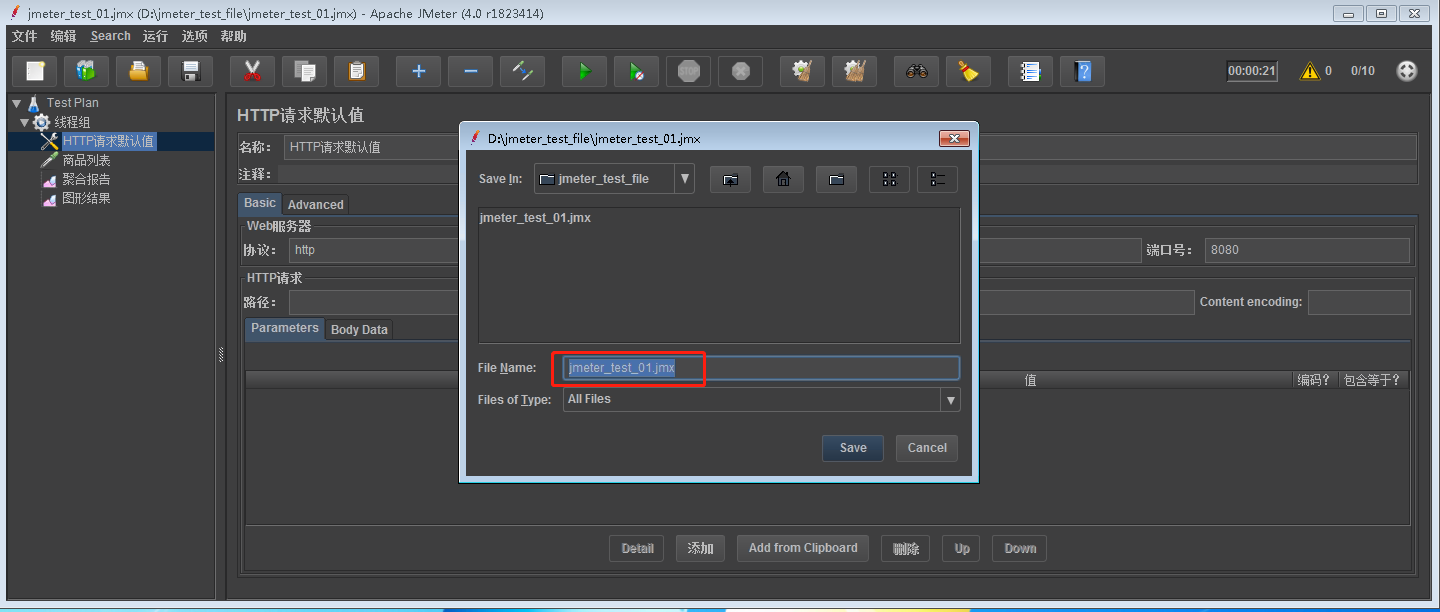
最新样本：最新样本响应时间，表示服务器响应最后一个请求的时间。（单位：毫秒）

偏离：服务器响应时间变化、离散程度测量值的大小，或者，换句话说，就是数据的分布。

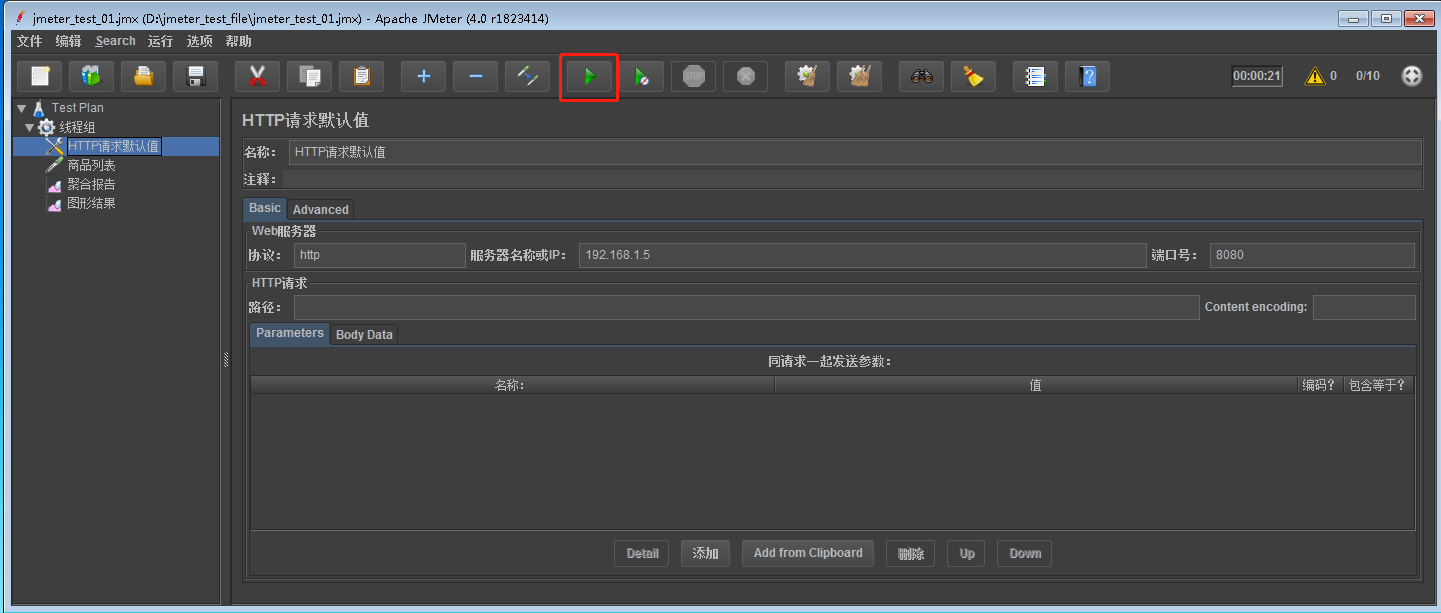


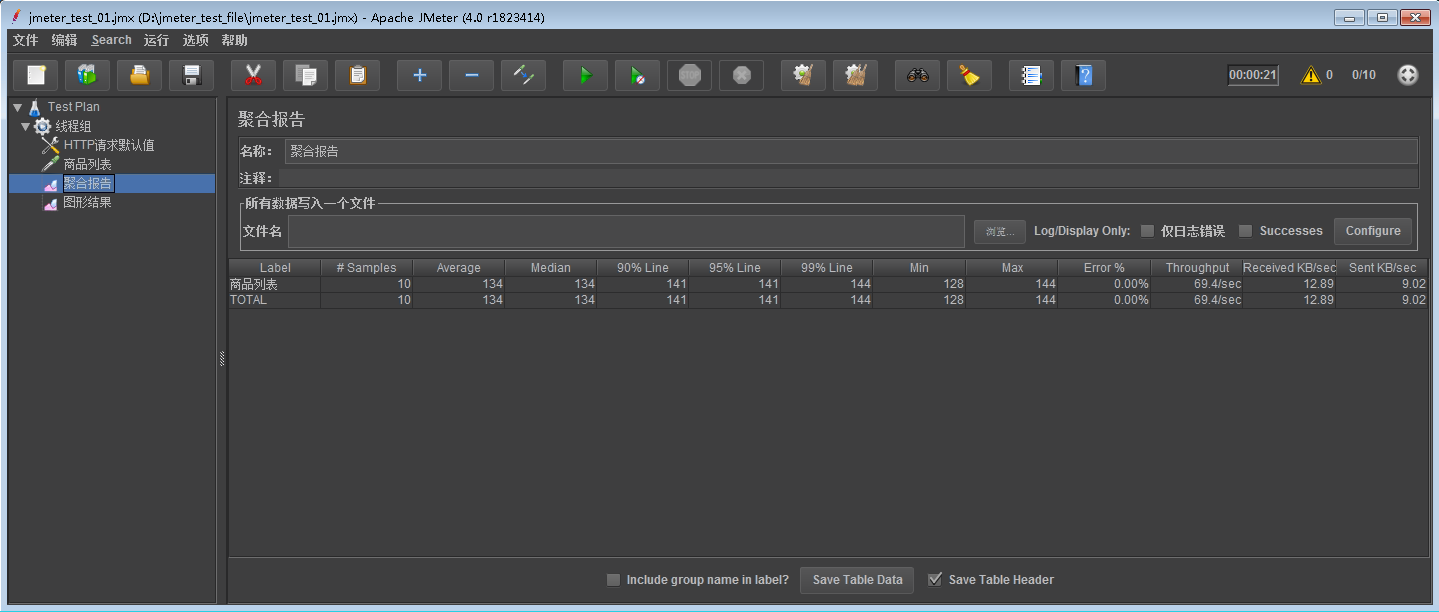
#### 步骤五：保存当前jmeter测试计划，并执行测试计划，查看测试报告，分析测试报告结果

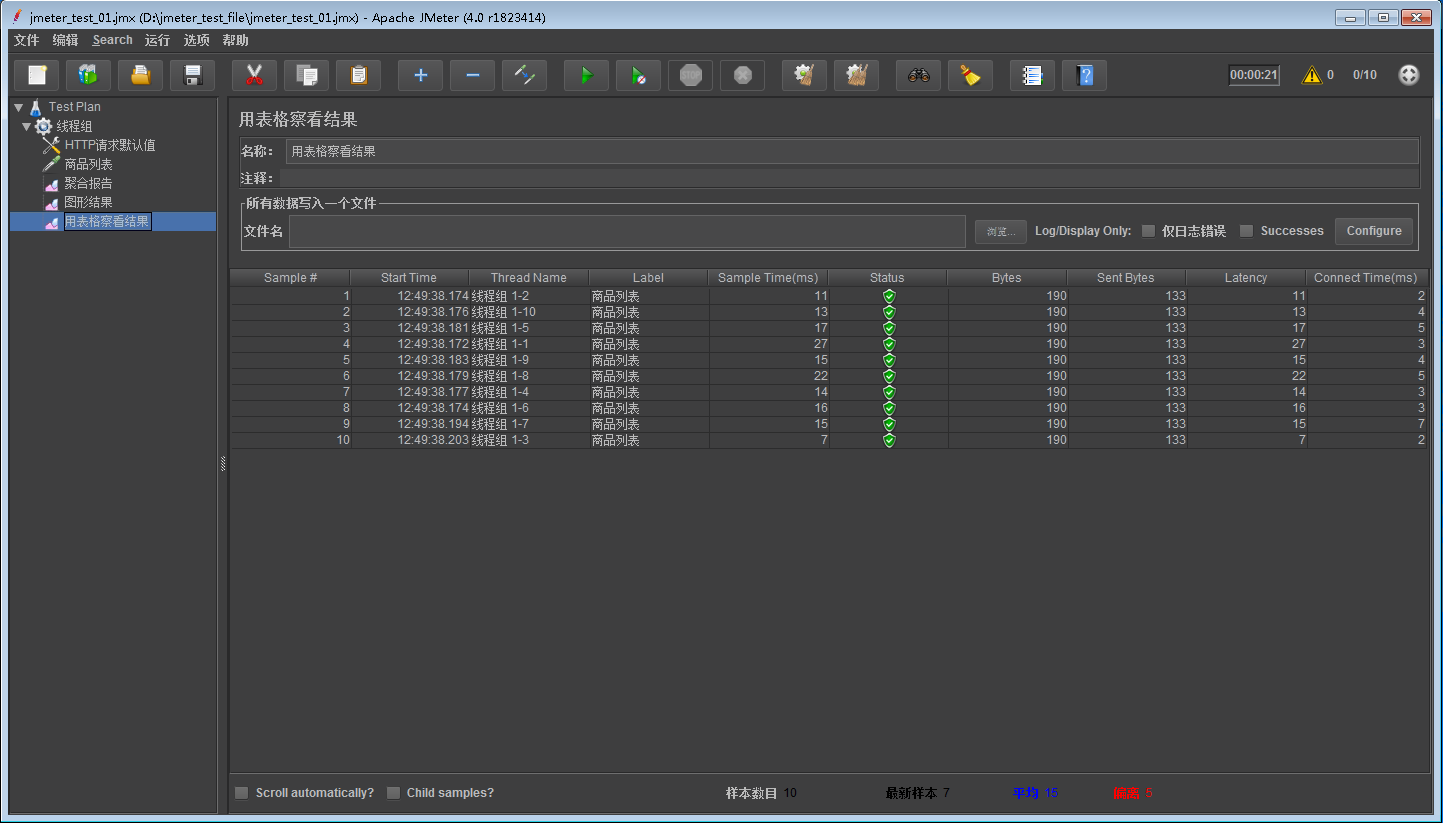
使用快捷键Ctrl+s将当前测试计划保存



保存完成后，选择HTTP请求默认值，点击绿色的启动按钮，对测试计划指定的接口进行压测以及分析报告输出。

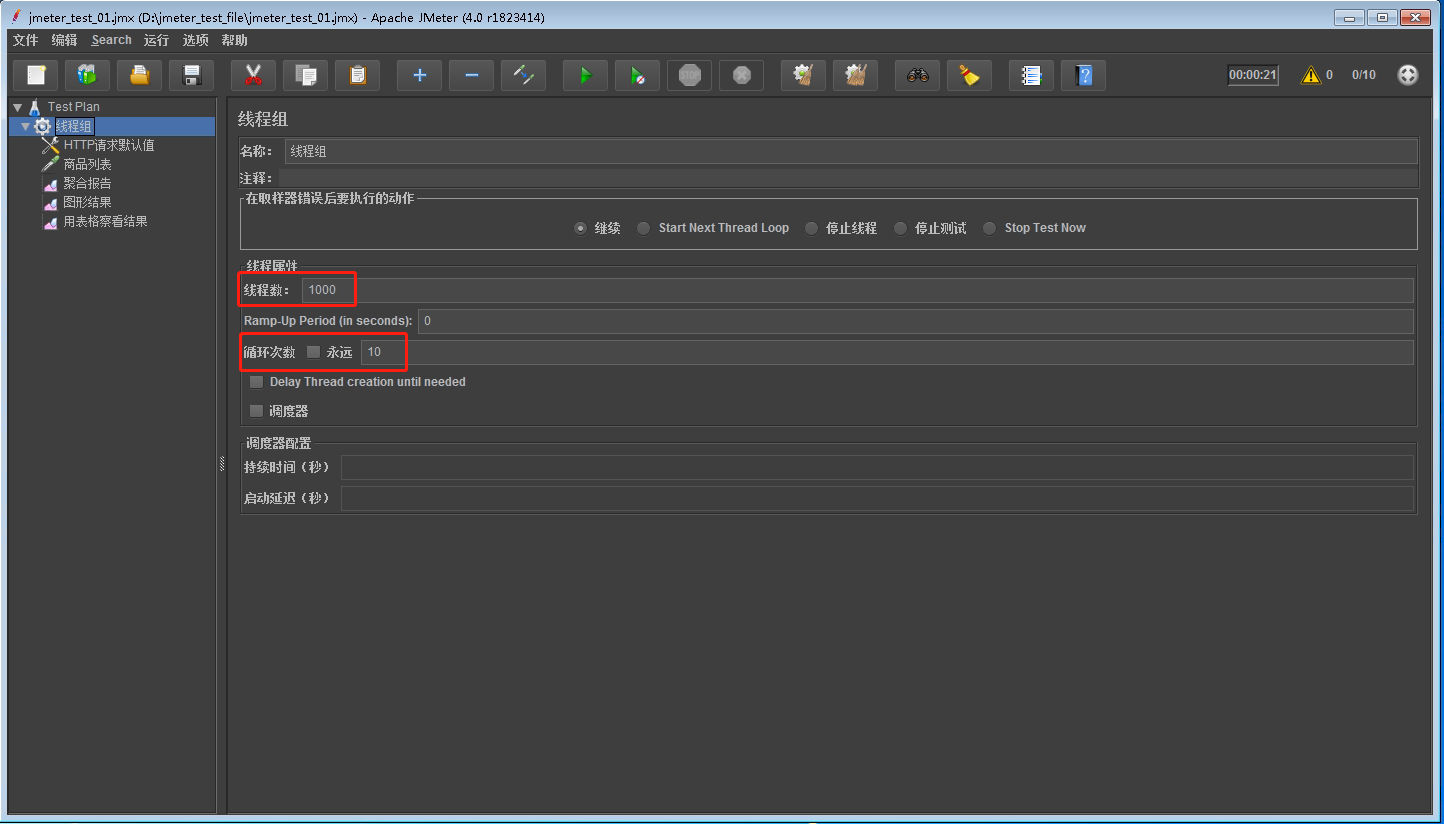


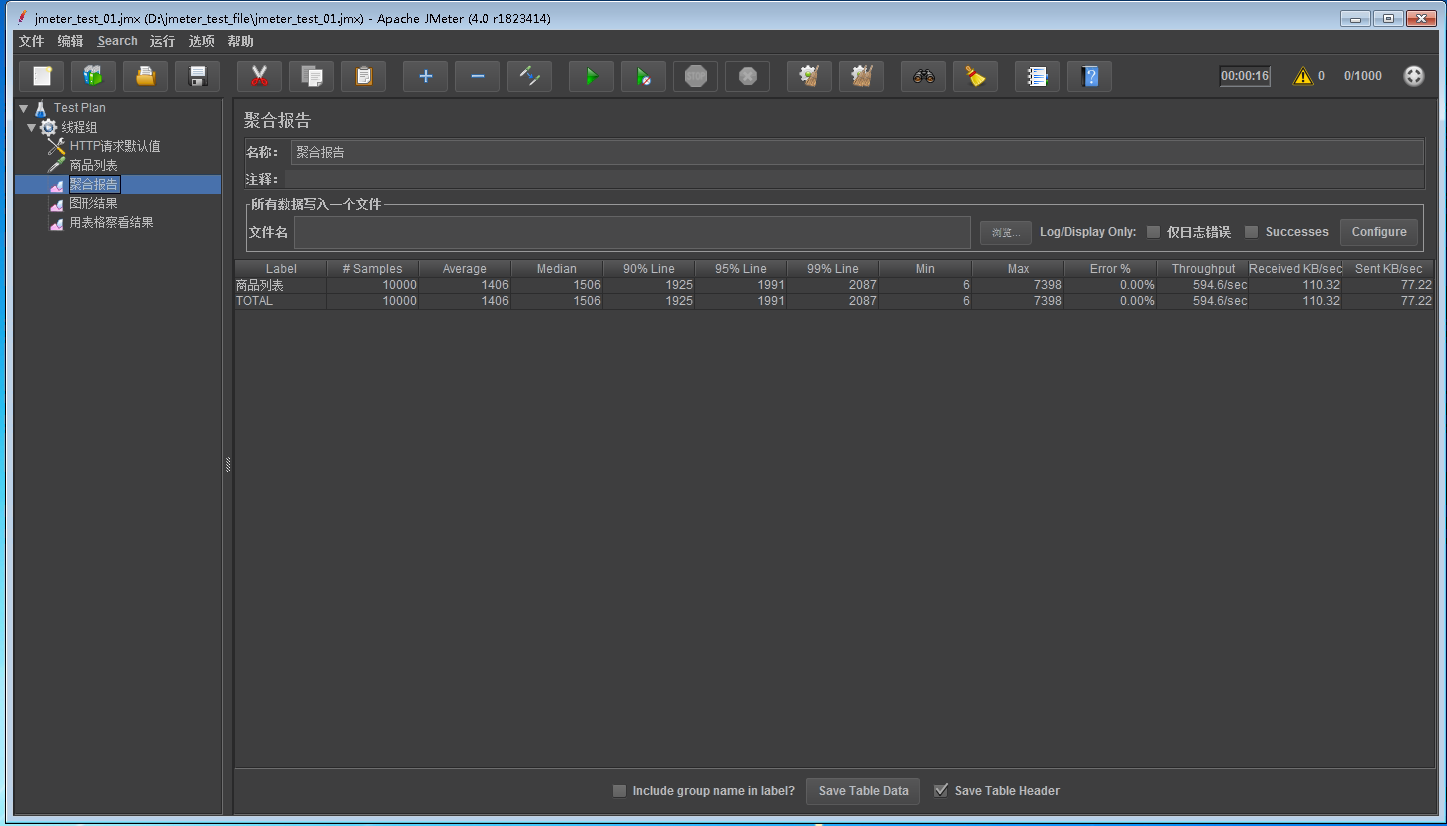


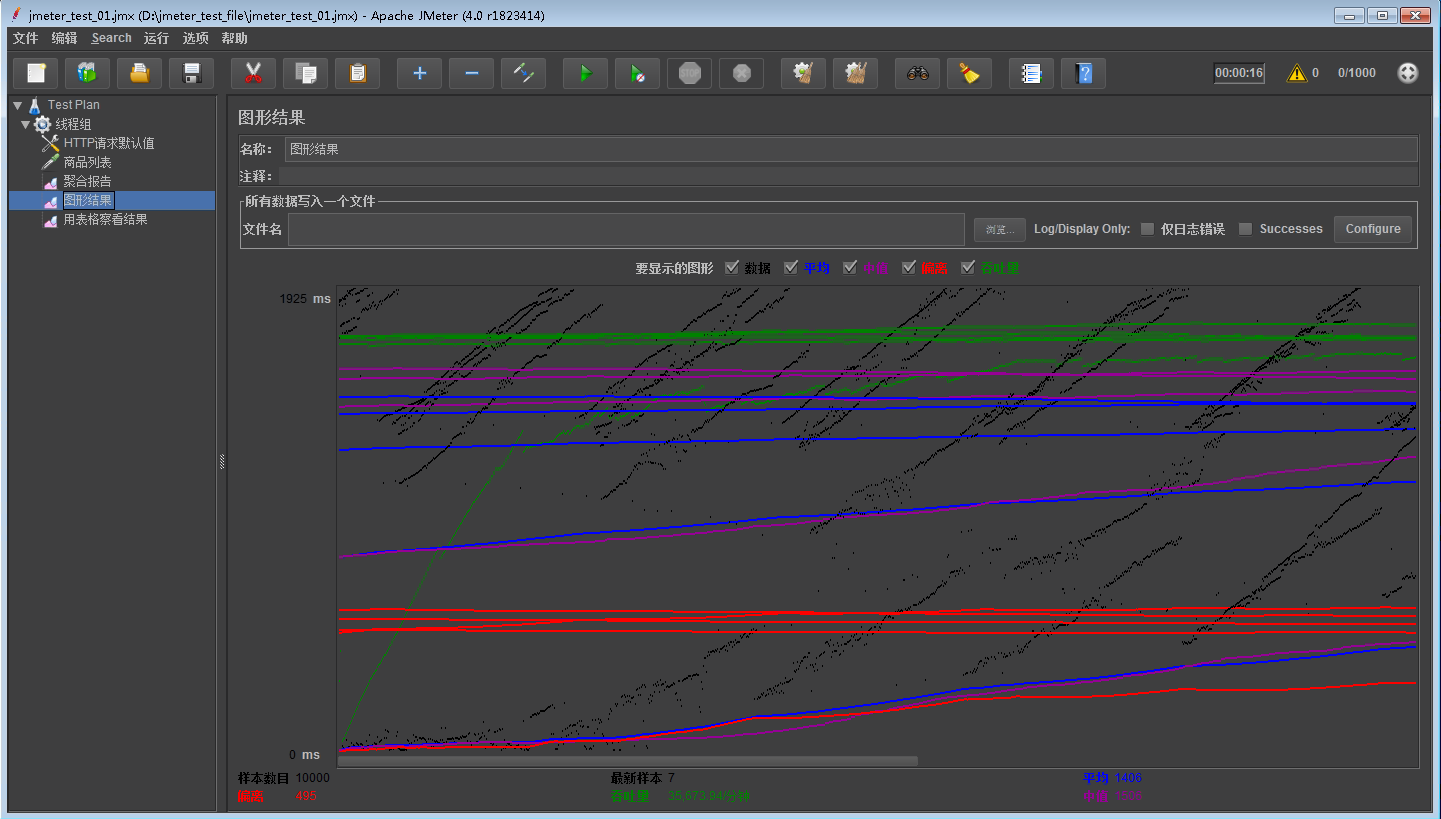


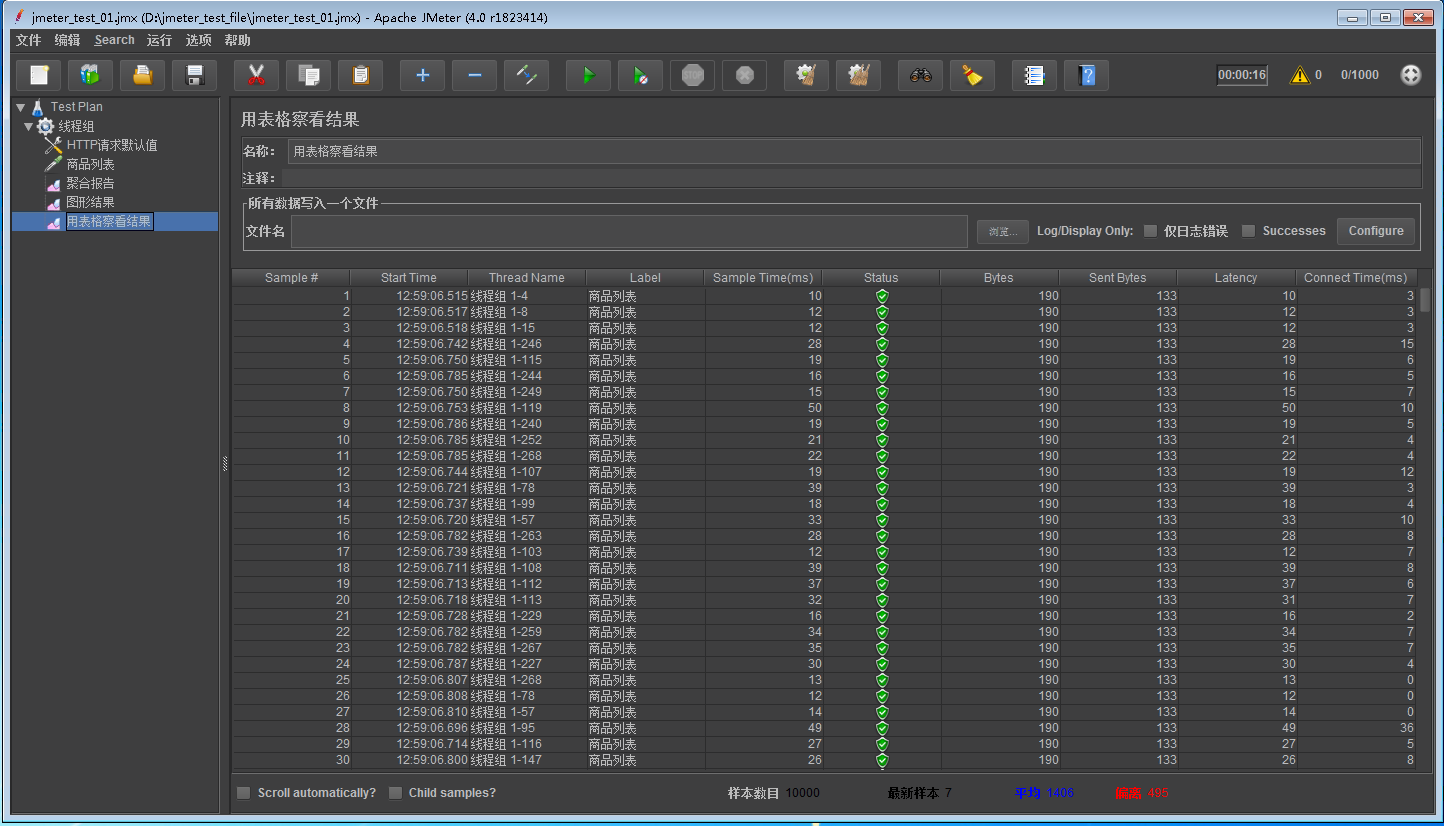
#### 步骤六：通过调整线程组中的属性来对指定接口压测，观察压测报告并分析并发量和峰值

通过提高线程组中的线程数来对指定接口进行压测，从而观察到接口能承受的QPS和TPS，实现jmeter压测的效果。





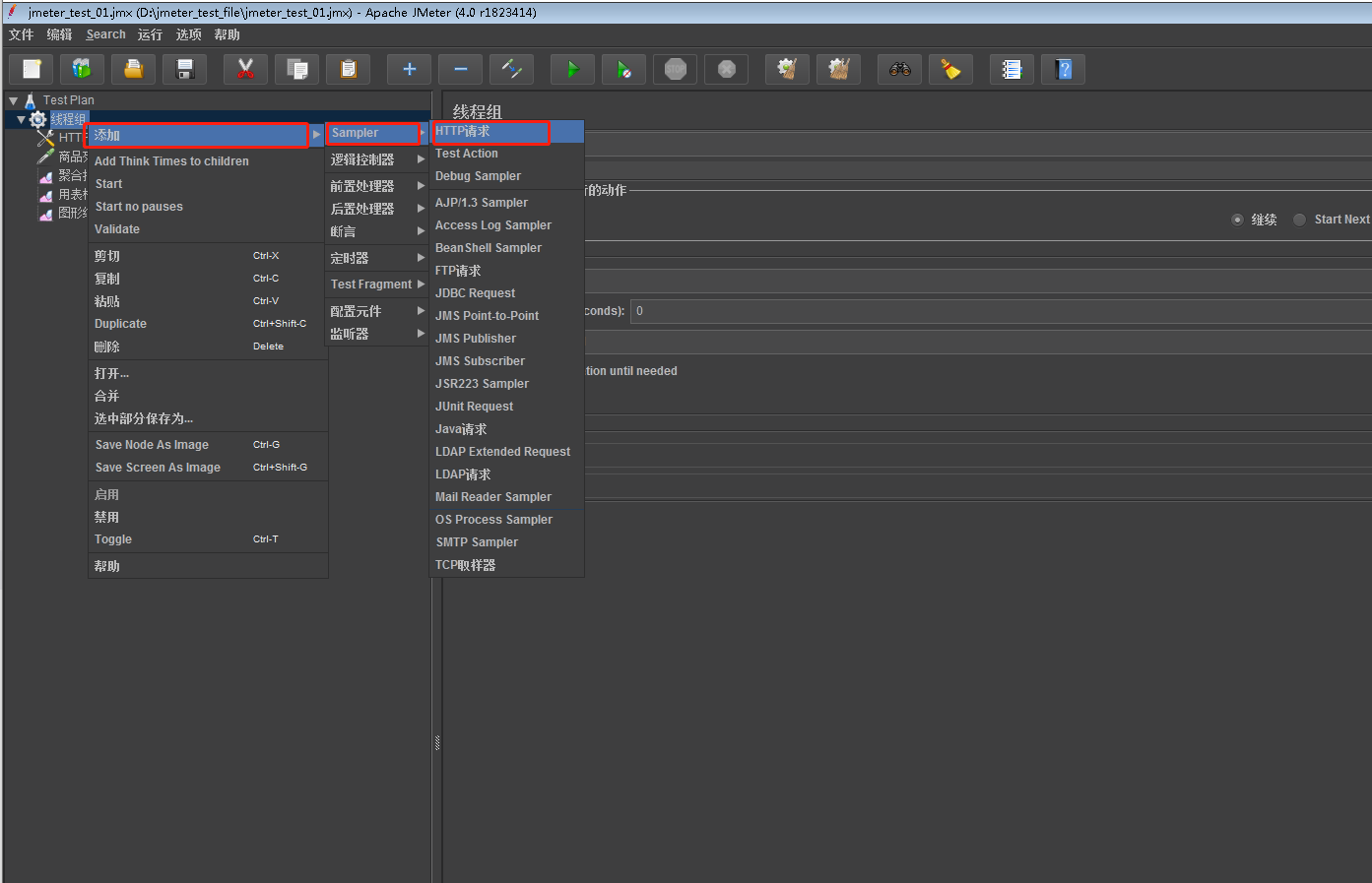




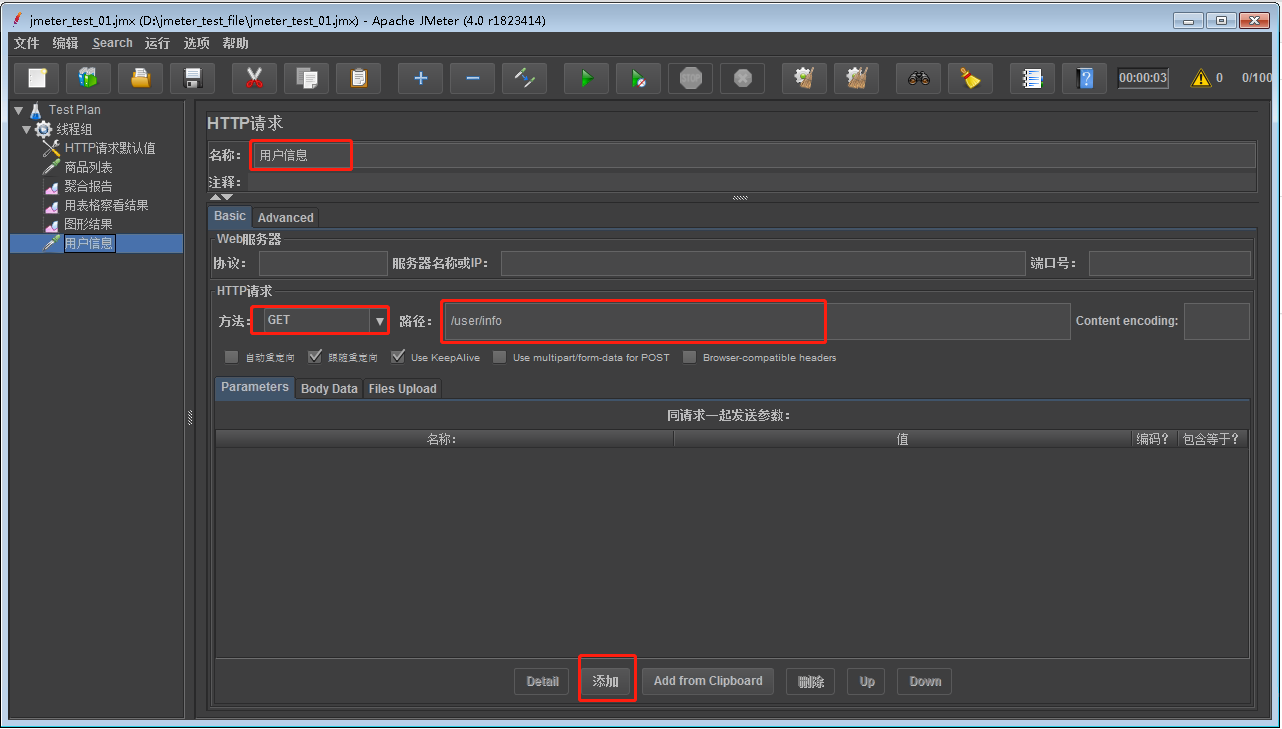
#### 步骤七：同一个测试计划下，通过禁用启用HTTP请求对不同接口进行压测，且互不干扰（带入参的HTTP请求压测）

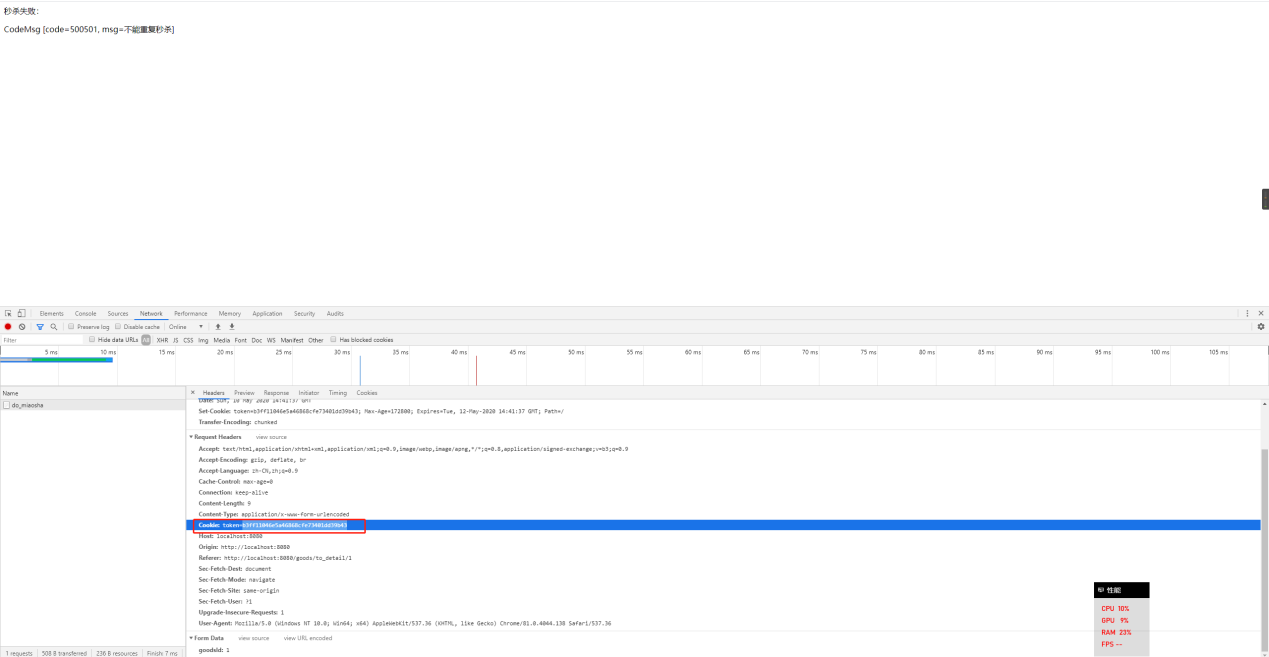
选择步骤一新建的线程组->右键->添加->Sampler->HTTP请求

新建获取用户信息HTTP请求，访问获取用户信息接口

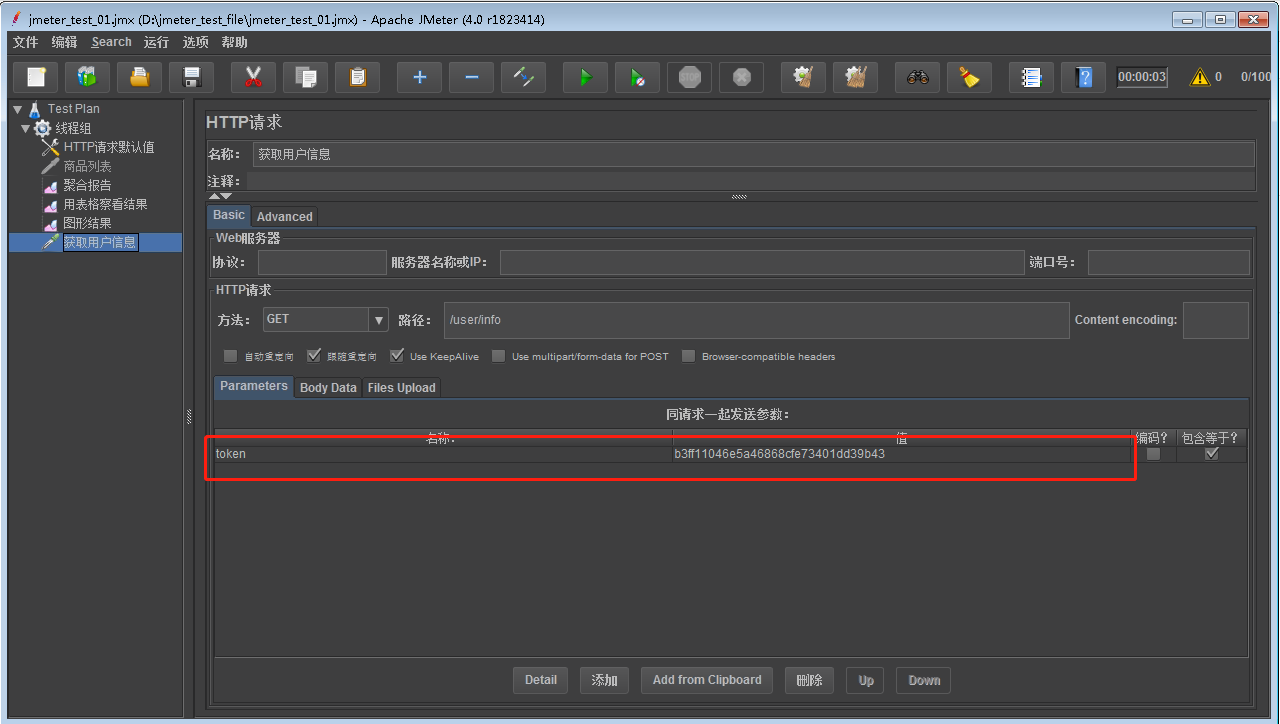


填写名称、方法和路径，然后点击添加按钮，添加请求接口的参数信息





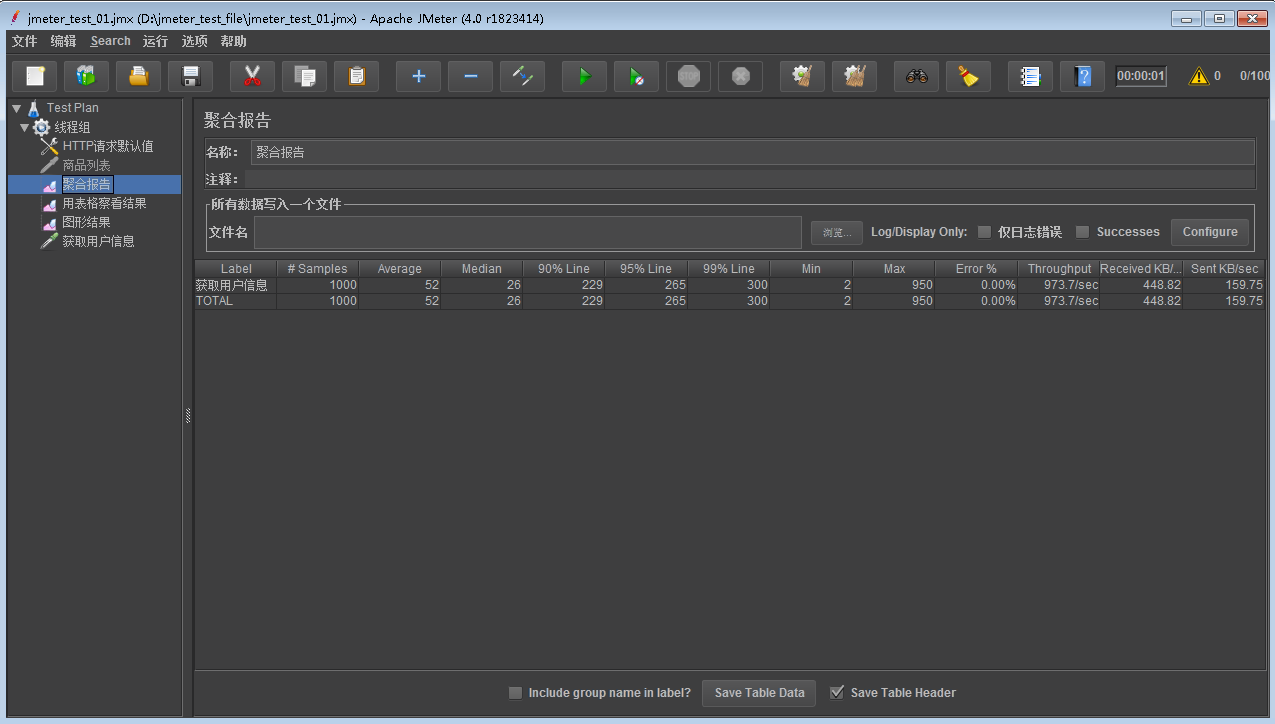
添加token请求入参



选择HTTP请求默认值，点击绿色开始按钮压测，结果如下：

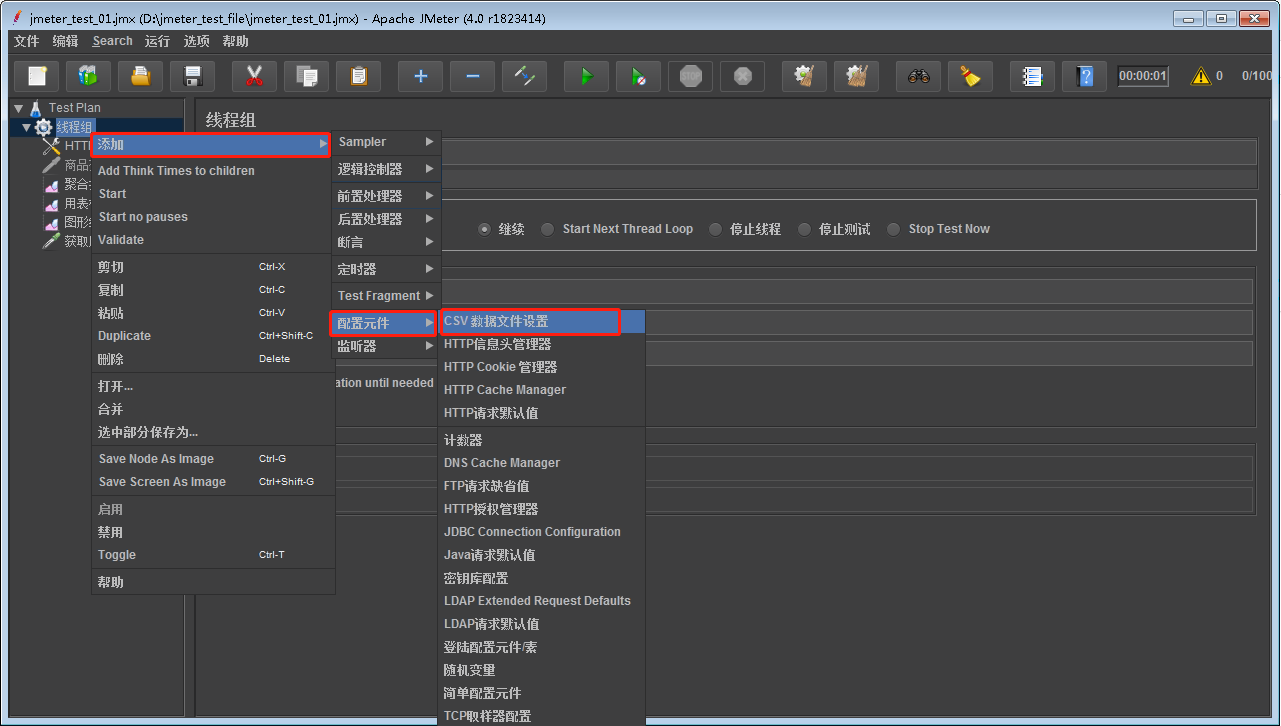


根据如上图结果优化项目后的压测结果如下图：



由于这里使用的token值是一个默认且一样的值，表示的是同一个用户，假如需要模拟多个不同用户进行压测，我们需要使用CSV Data Set Config（CSV数据文件设置），配置和操作如下：

选择线程组右键->添加->配置元件-> CSV Data Set Config（CSV数据文件设置）



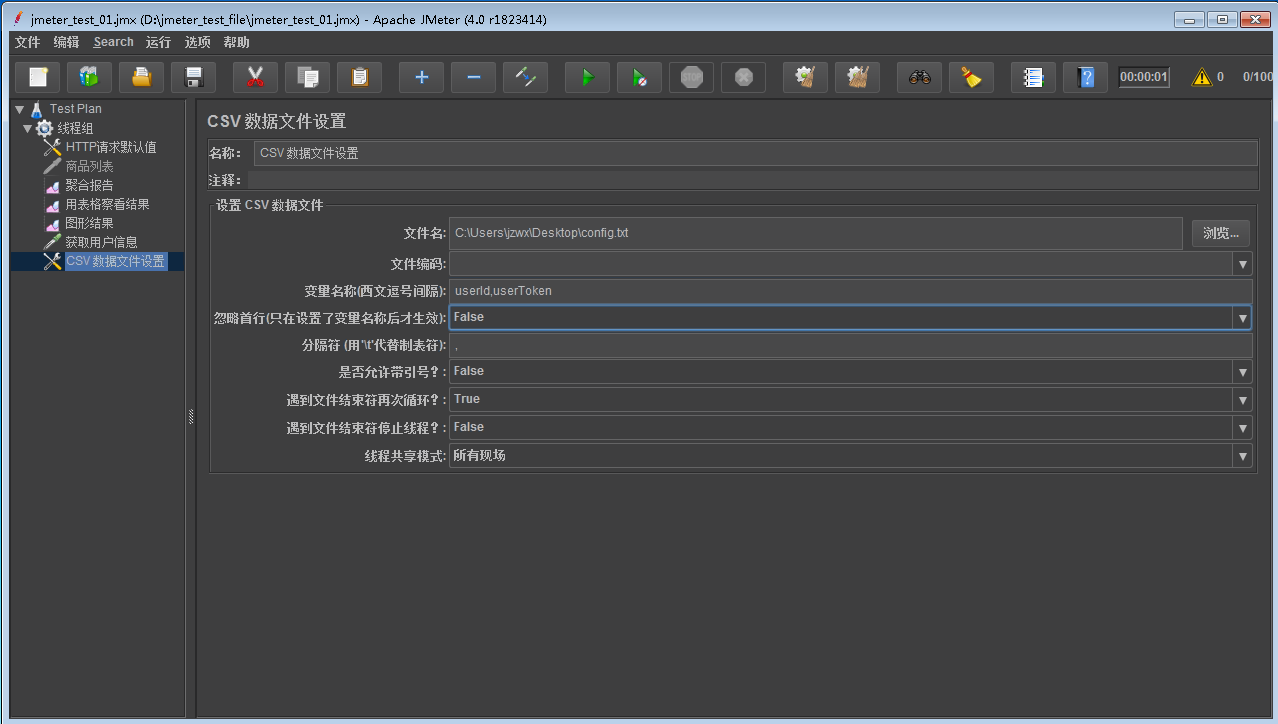


新建config.txt文件

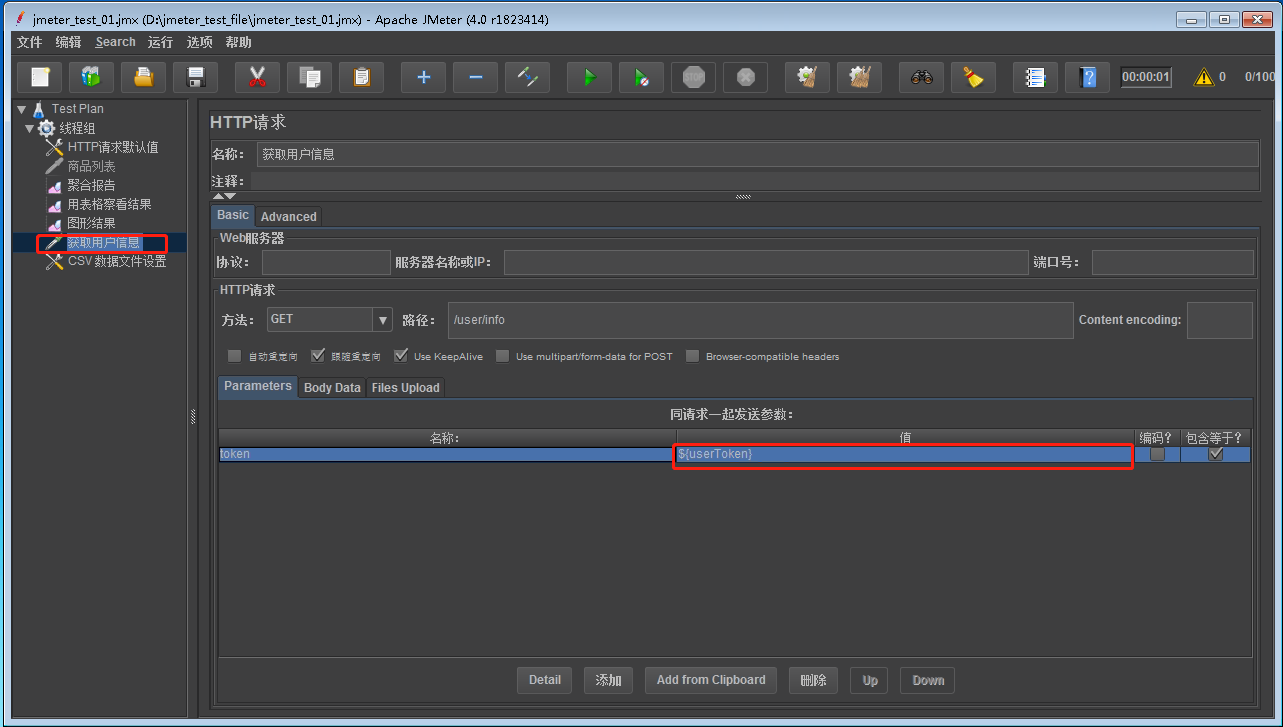
模板文件

案例中使用的文件

CSV Data Set Config（CSV数据文件设置）配置信息如下：

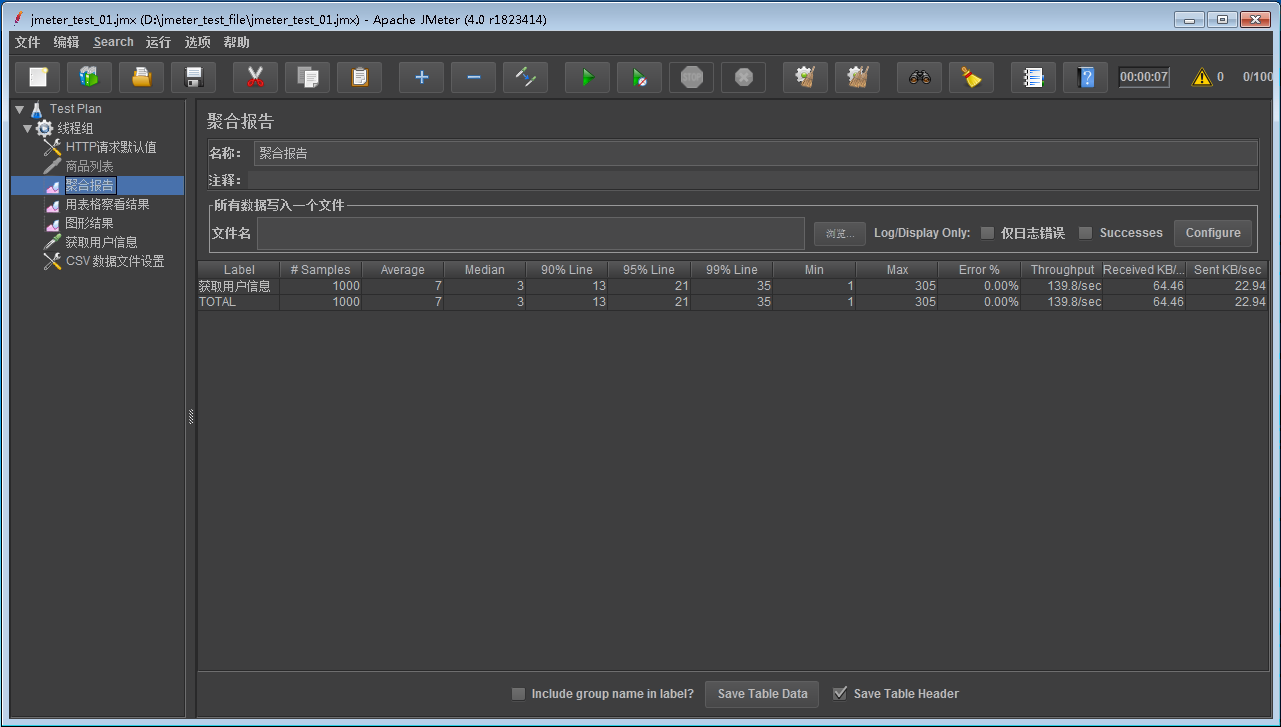


文件中入参信息（userId，userToken）引用到获取用户信息HTTP请求中做请求入参使用，应做如下设置：

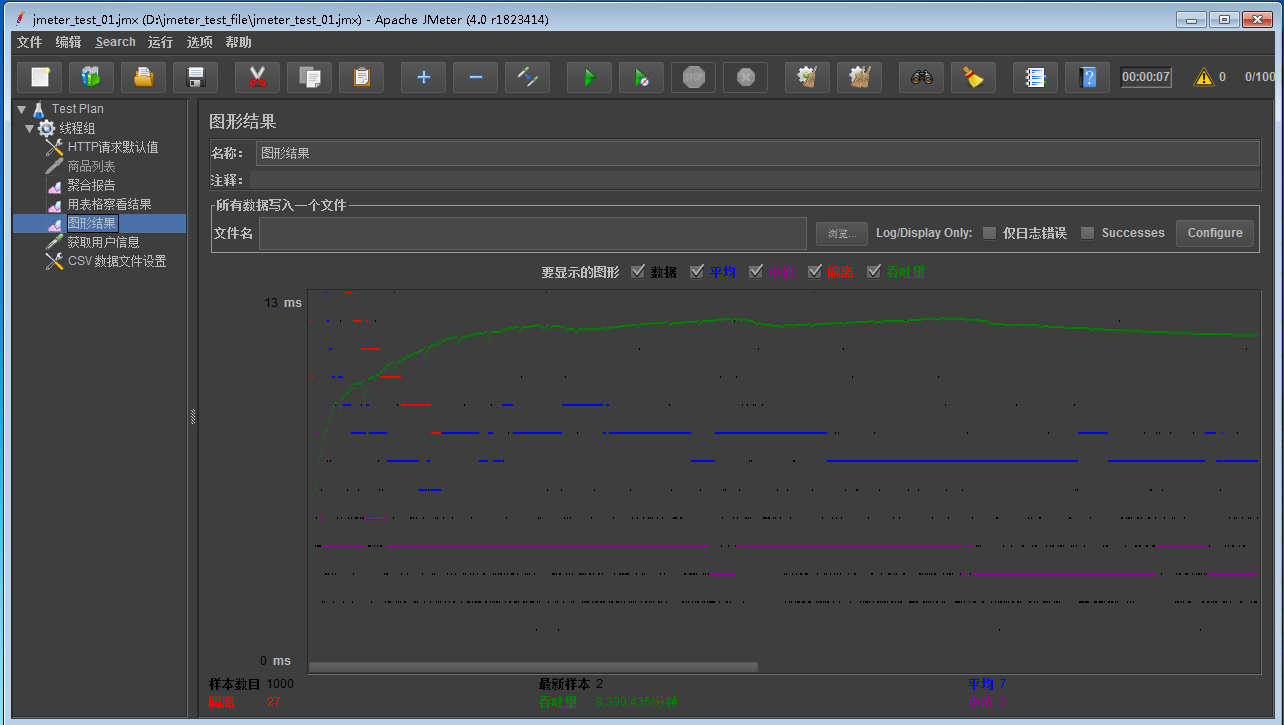


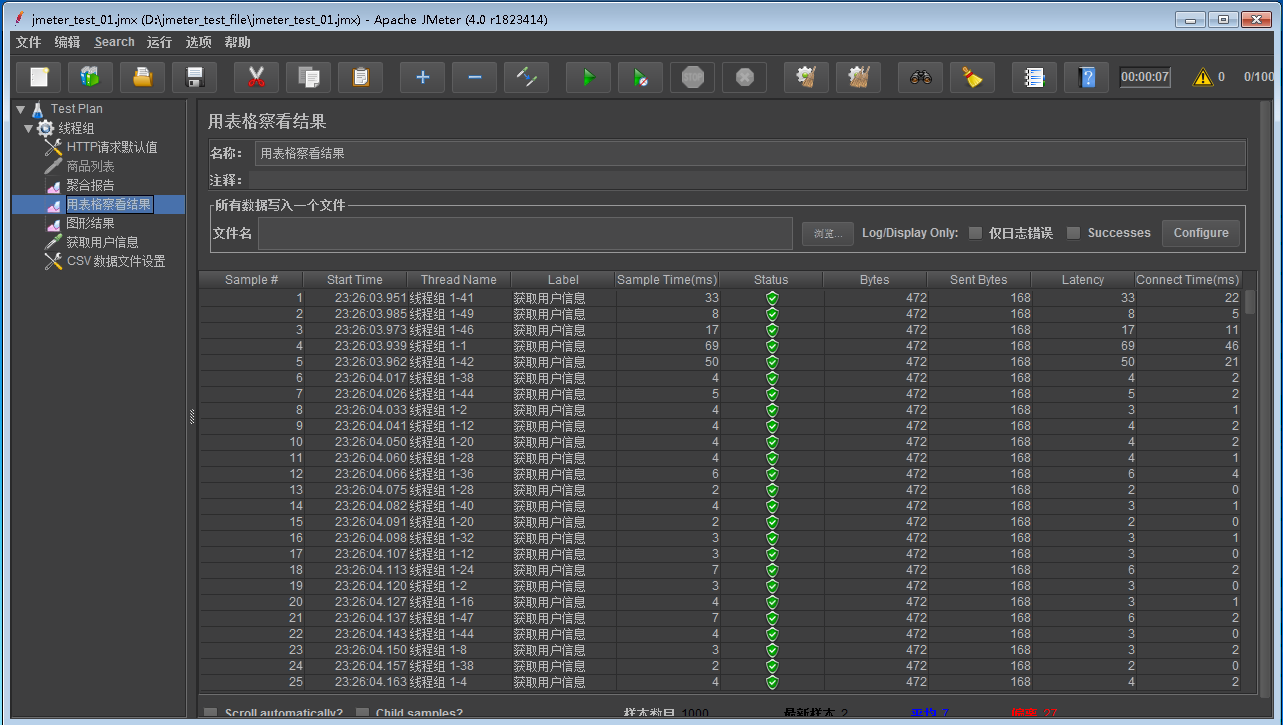
使用${userToken}设置成token，做压测时会${userToken}会同config.txt文件中读取对应的userToken作为获取用户信息请求入参的token参数传给后端进行接口请求获取数据，这样一个情况。

以下是通过CSV Data Set Config（CSV数据文件设置）再次对获取用户信息接口进行压测结果如下：



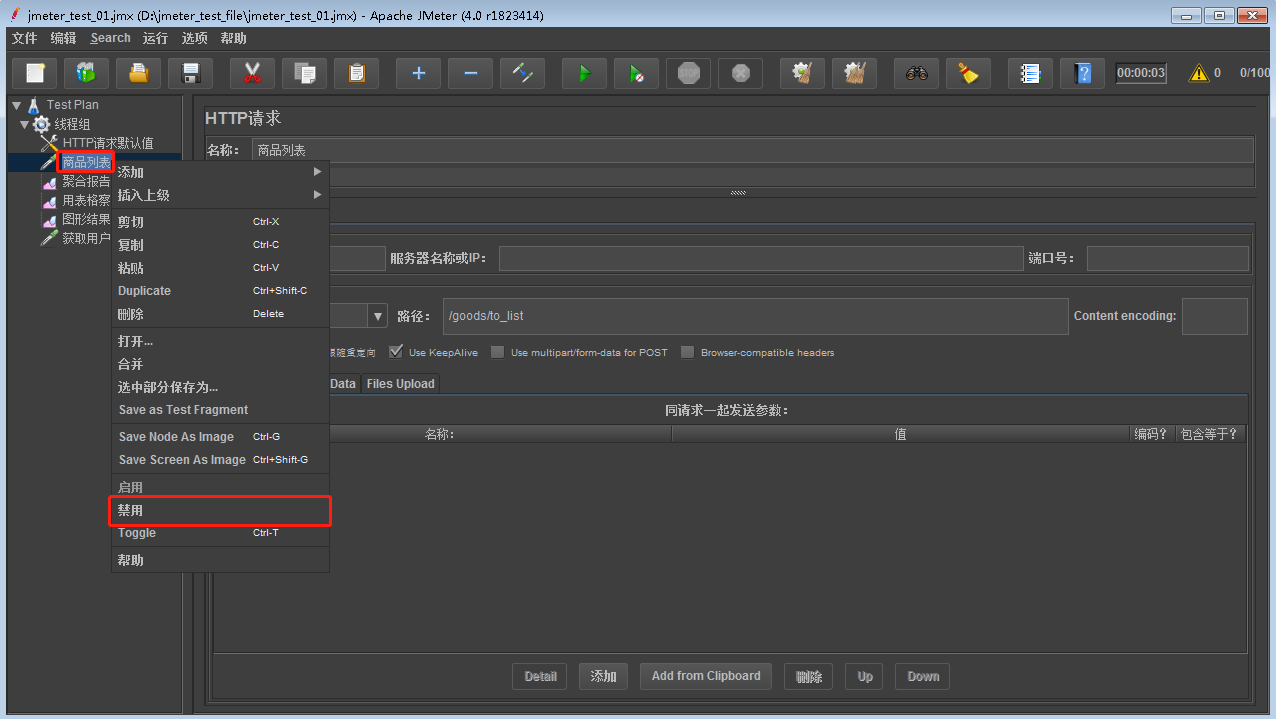
以上是模拟不同用户请求获取用户信息接口，实际是通过新建CSV Data Set Config（CSV数据文件设置）文件，然后引用配置文件进行模拟操作，这样就行了。

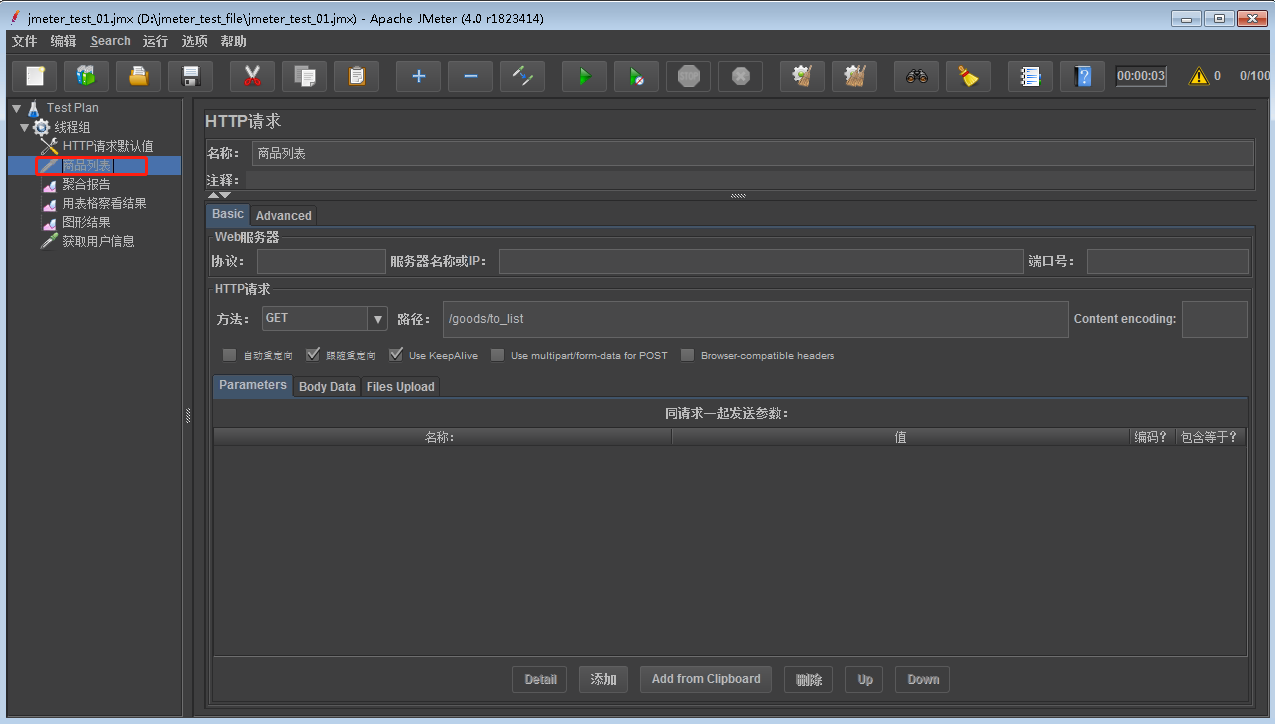




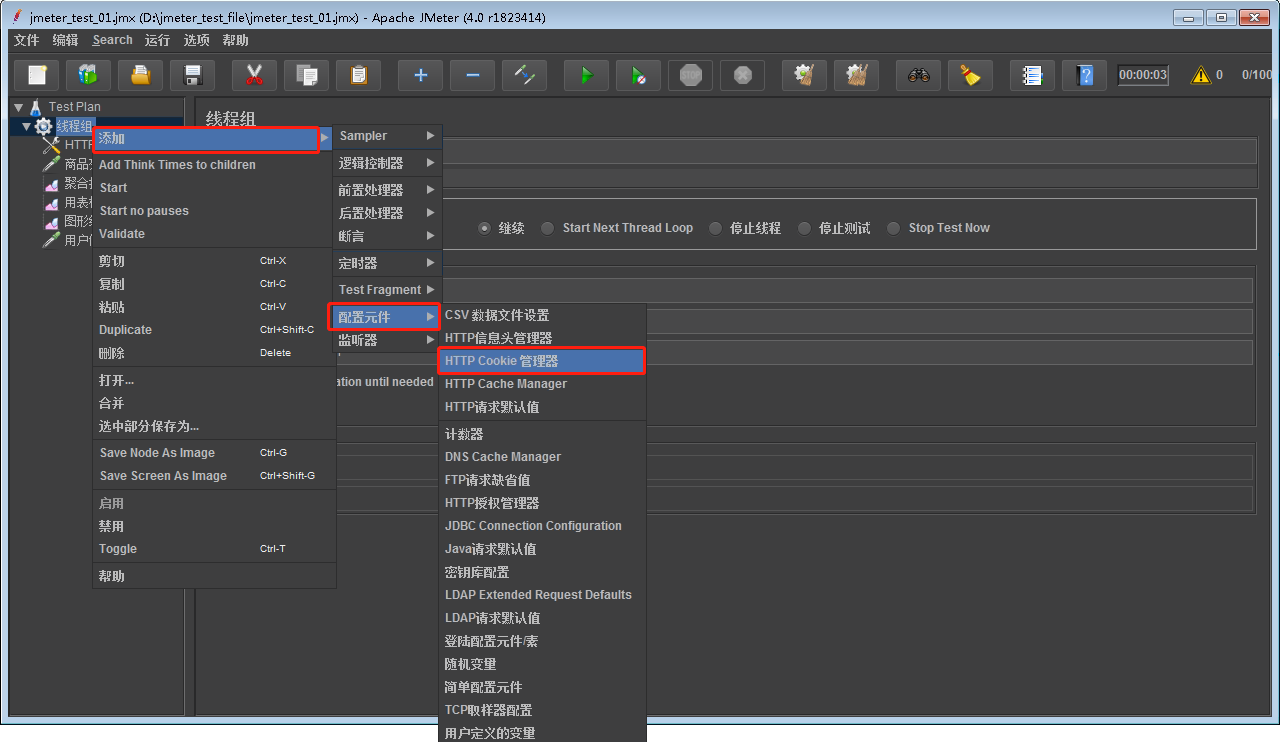
禁用商品列表HTTP请求，本次测试计划只对获取用户信息做压测，禁用之后左侧的商品列表菜单变灰色

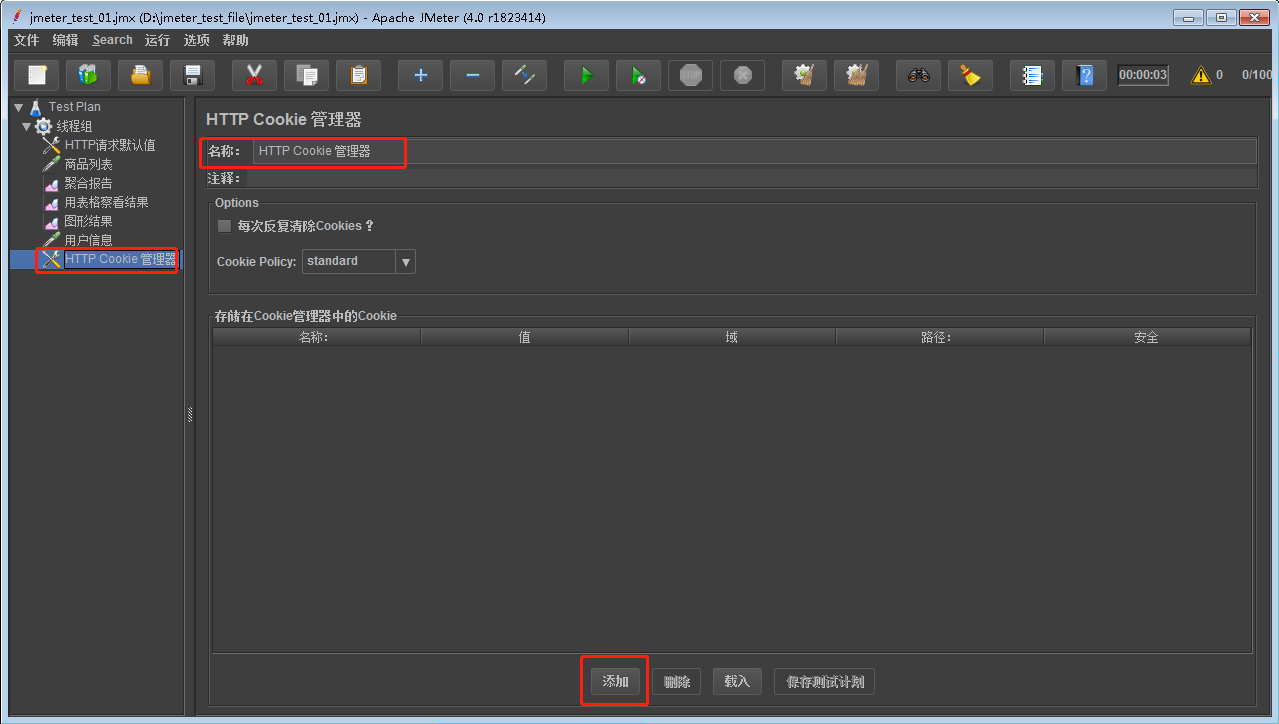
选择商品列表菜单右键->禁用





Jmeter同时也支持cookie传请求入参信息（本次暂时不予介绍）





参考博客：

<https://blog.csdn.net/qq_39720249/article/details/84313530>

<https://www.cnblogs.com/atwanli/articles/4908475.html>

## 自定义变量模拟多用户（带参数请求的压测）

注意：此部门内容在jmeter入门的步骤七已经介绍，这里就不再继续讲解

## JMeter命令行使用

### 一、jmeter命令行，执行大致过程

1.在windows上录好jmx

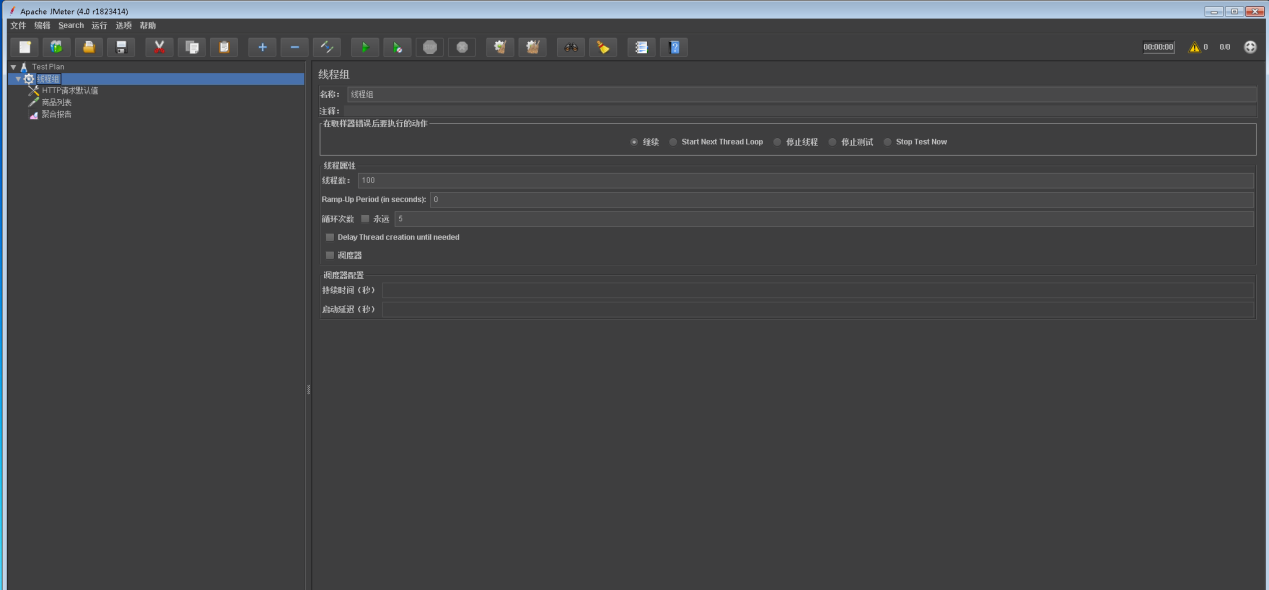
2.命令行：sh jmeter.sh -n -t xxx.jmx –l result.jtl (-n:不适用全局变量 -t:脚本 -l:输出结果到)

3.把result.jtl导入到jmeter

### 二、使用命令行压测项目实战

步骤一：jmeter中新创建一个测试计划

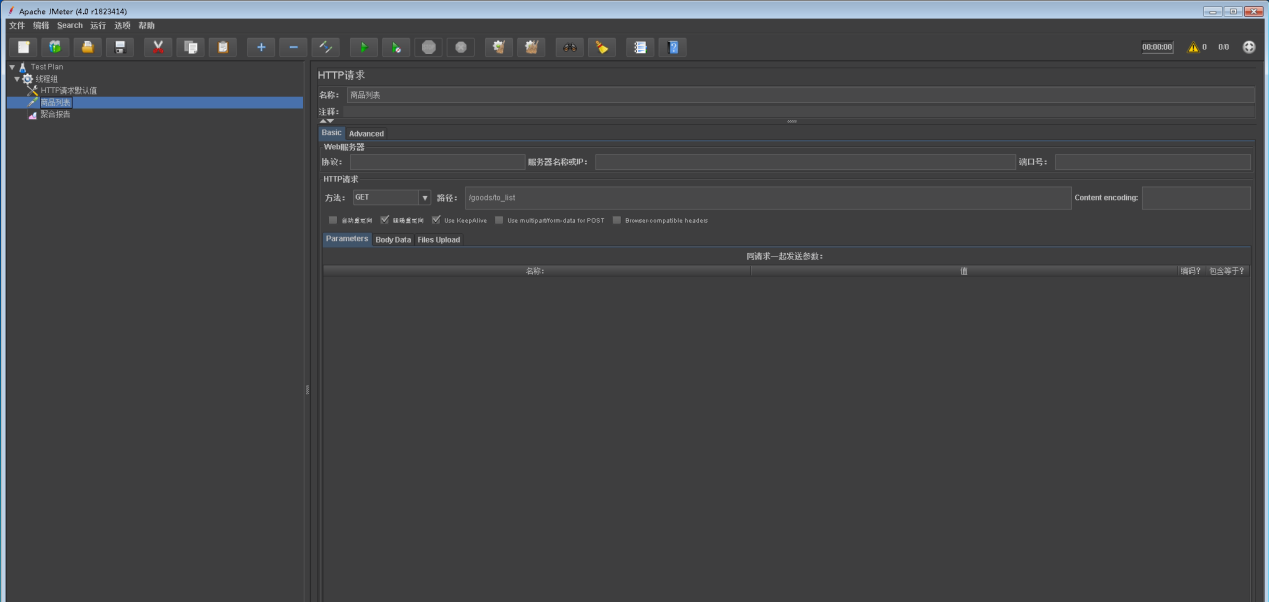
新建线程组，如图：



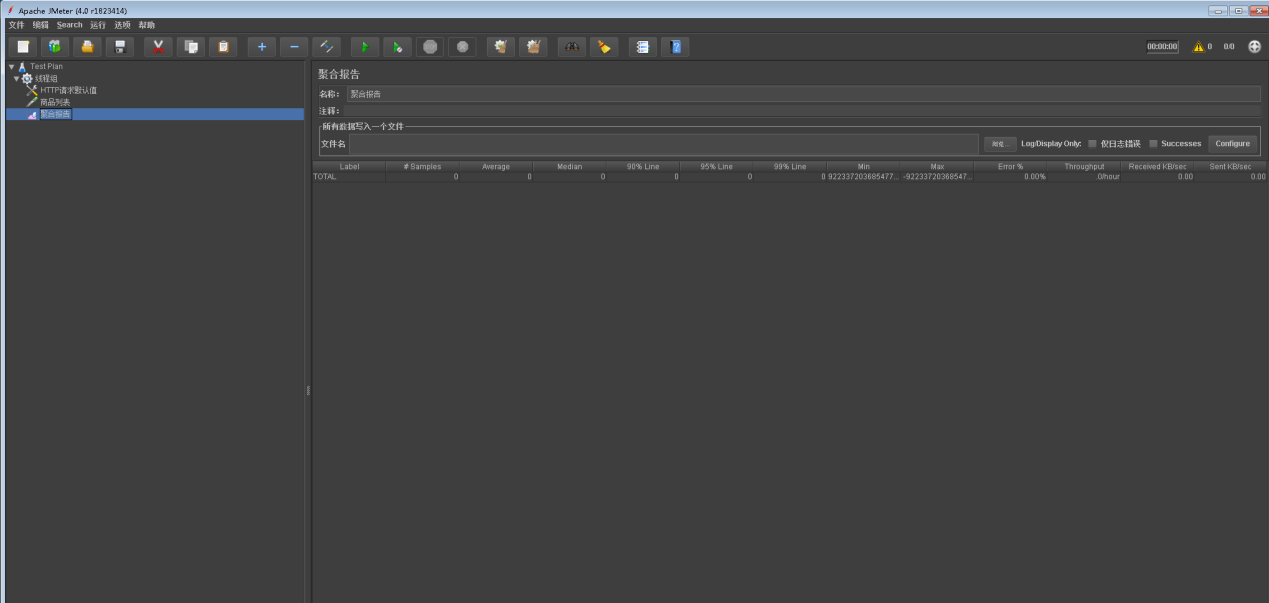
新建HTTP请求默认值，如图：



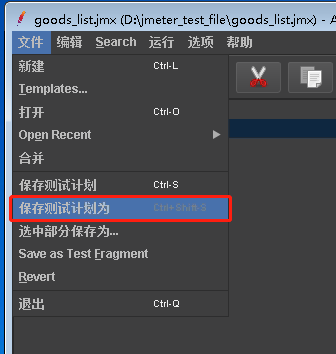
新建HTTP请求，如图：

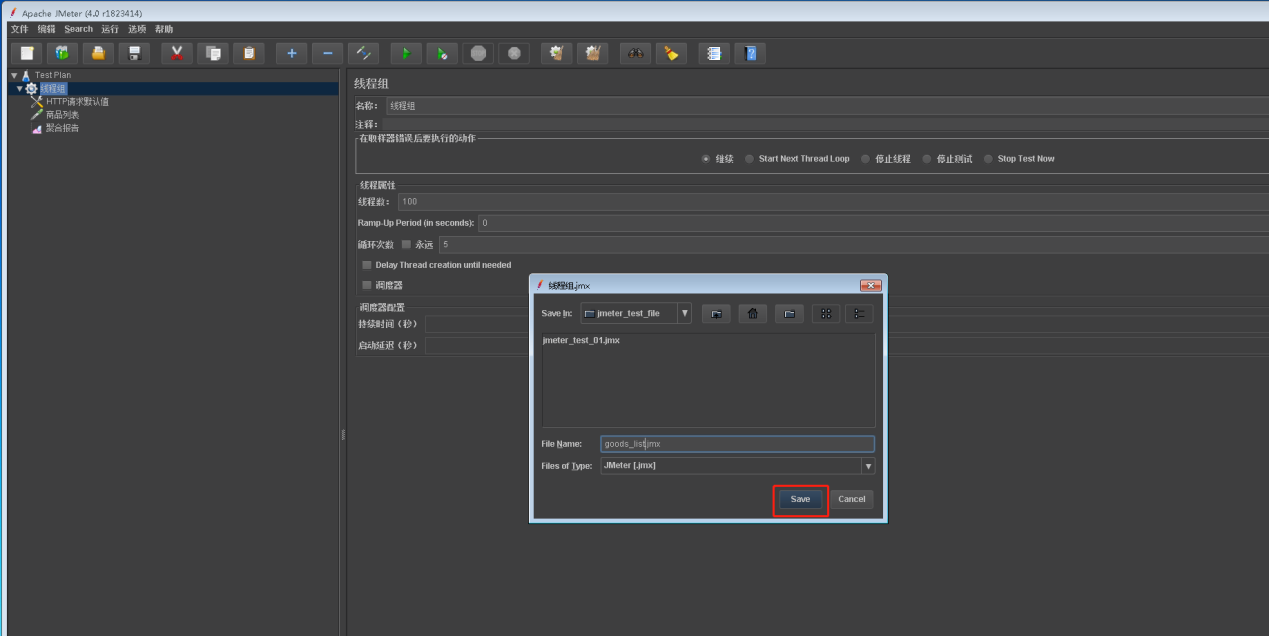


新建聚合报告，如图：



将新建好的测试计划保存起来命名为goods\_list.jmx，如图：





然后把生成的goods\_list.jmx文件上传到linux上，进入jmeter安装目录的bin目录下，goods\_list.jmx文件也放在bin目录，执行指令：

./apache-jmeter-3.3/bin/jmeter.sh -n -t goods\_list.jmx -l result.jtl

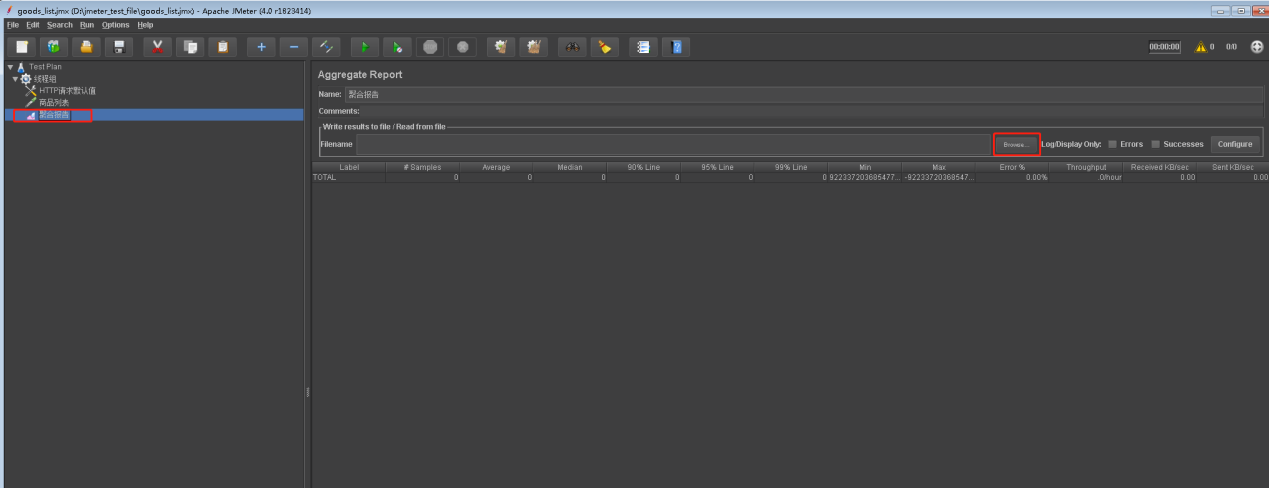
获得压测结果文件：result.jtl

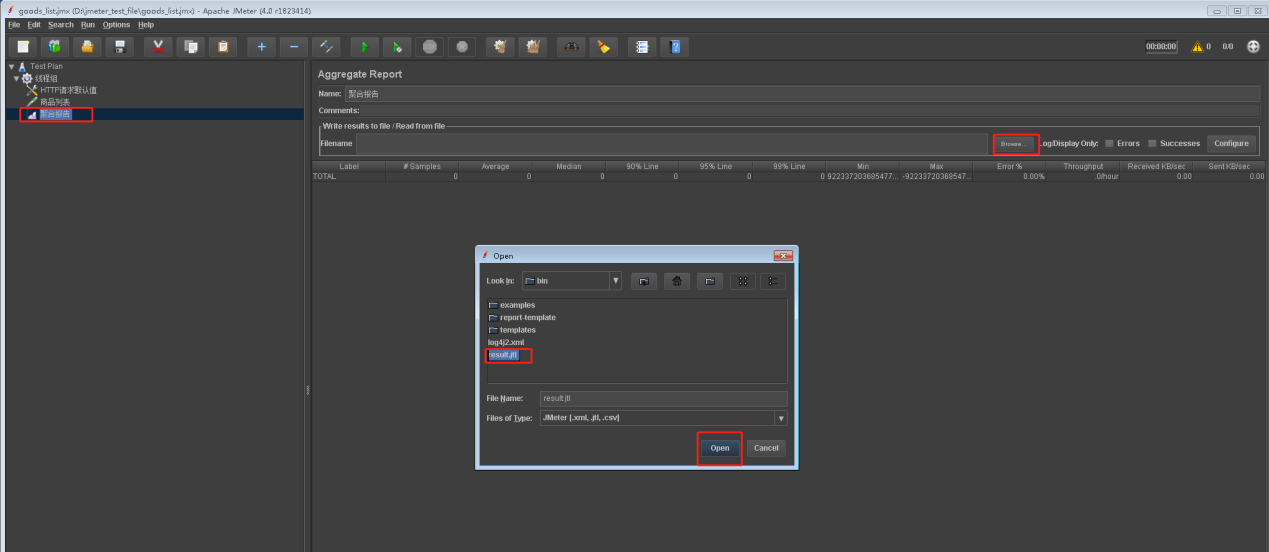
linux上传文件：rz

linux下载文件：sz

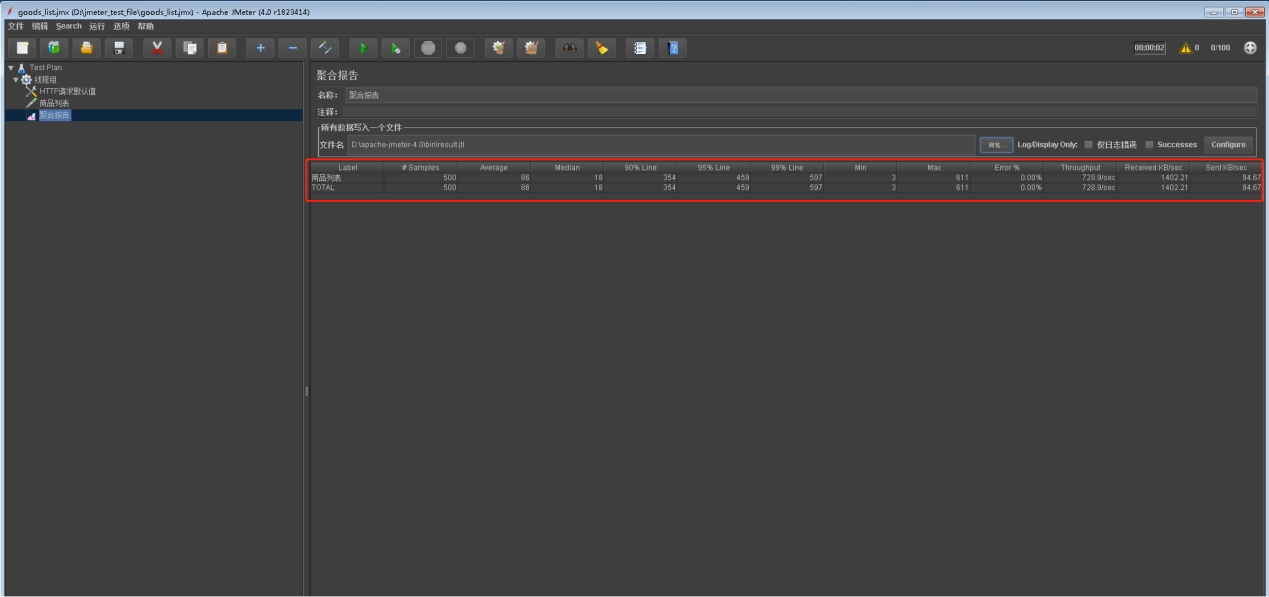
查看linux服务器内核数指令：cat /proc/cpuinfo | grep processor

将压测文件下载到windows下，进入jmeter打开goods\_list.jmx测试计划，选择聚合报告，选择linux下载下来的result.jtl文件查看测试结果，如下图：



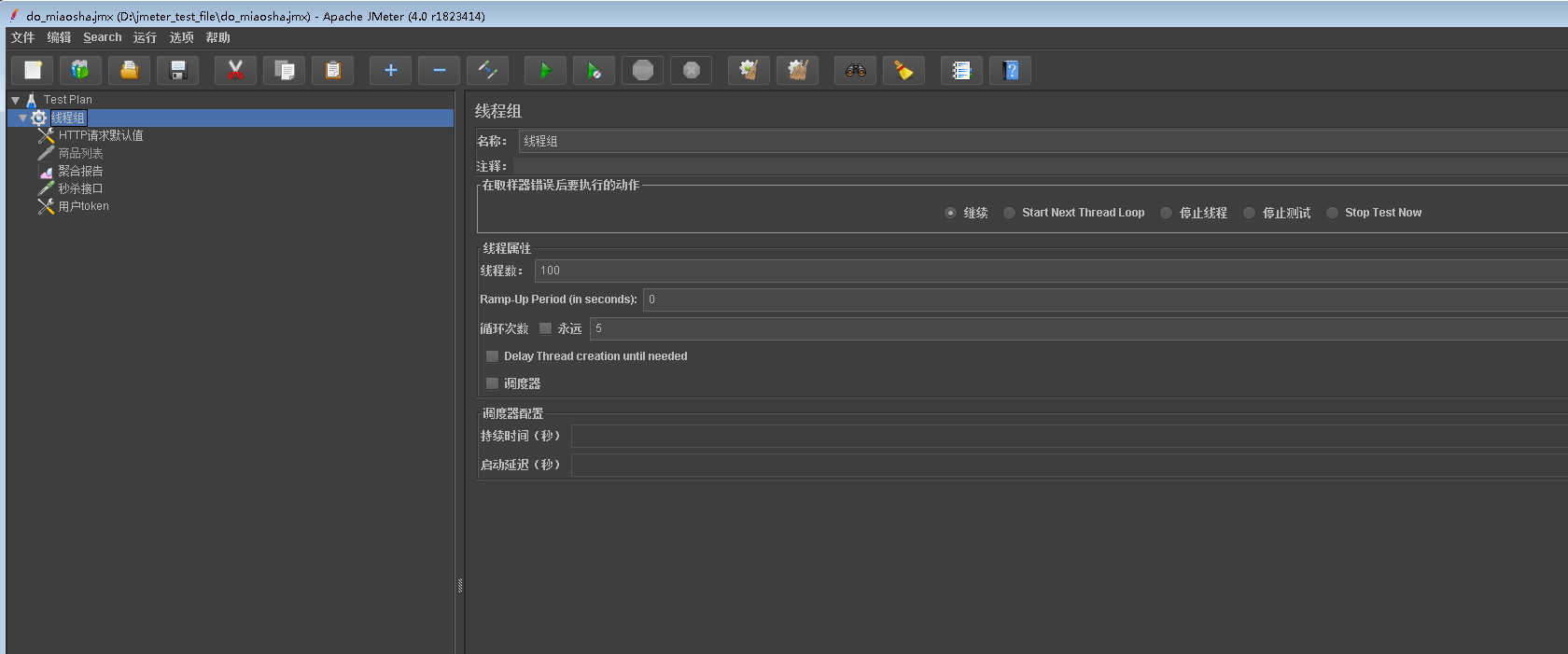


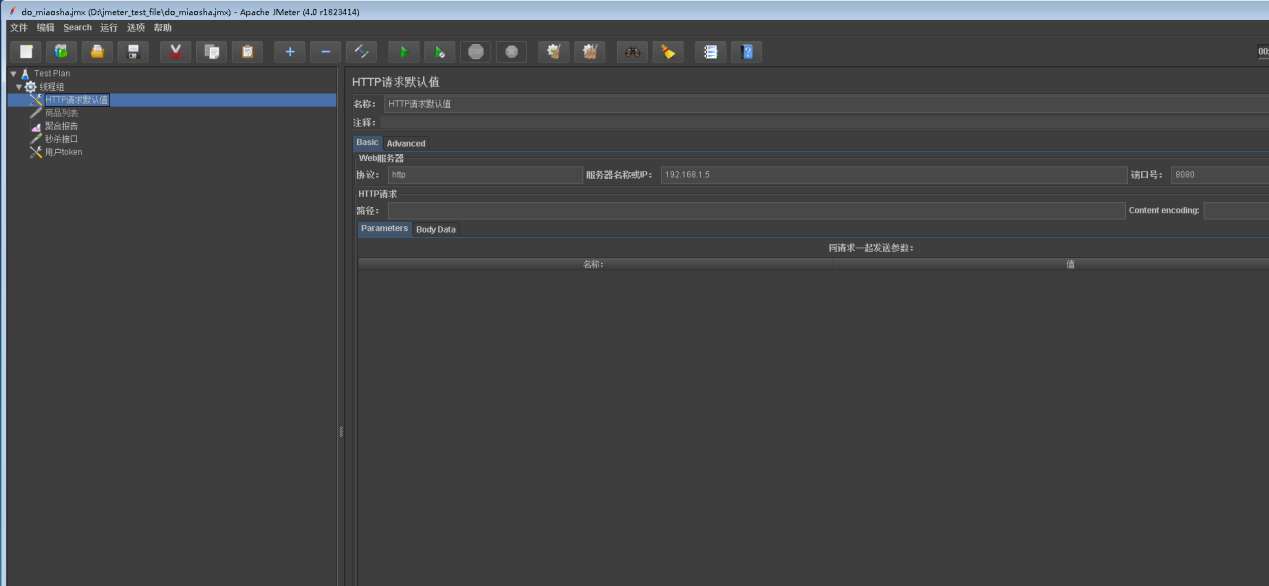
对压测结果进行查看，如图：

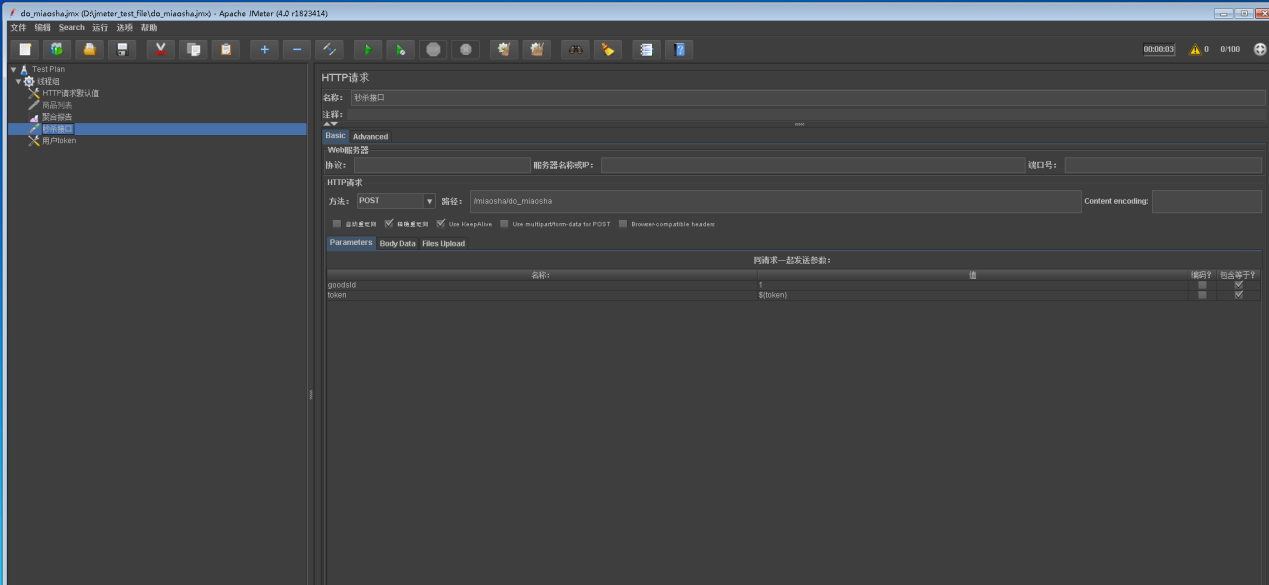


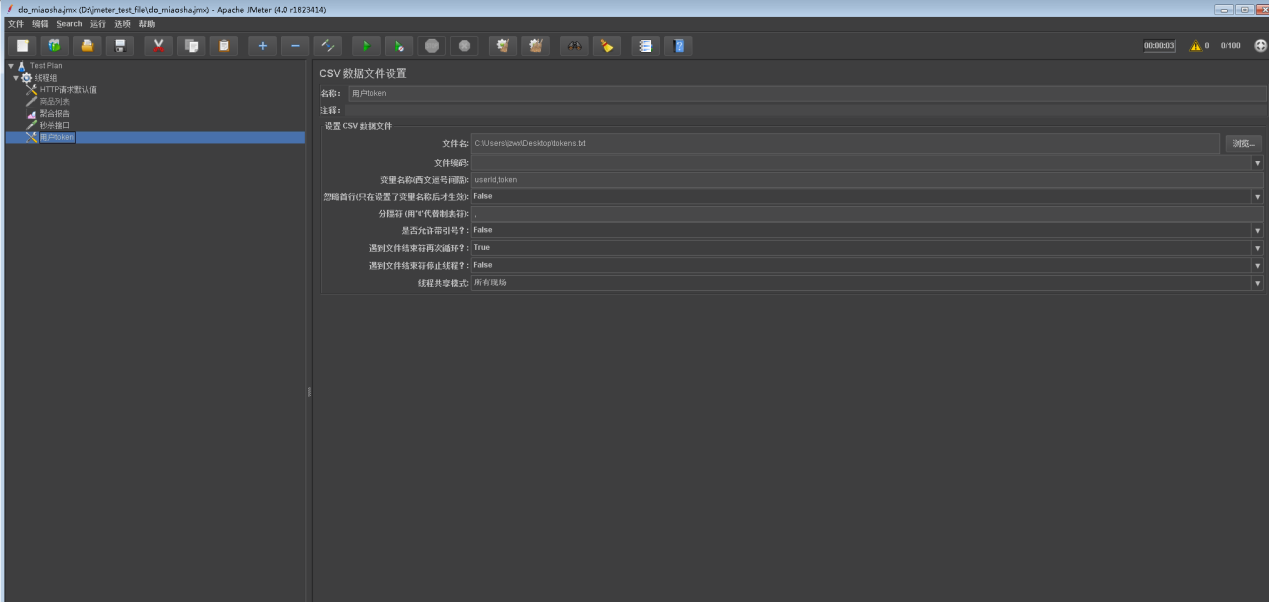
### 三、创建100个用户同时，同时产生100个并发，循环5次

创建测试计划，线程组，HTTP请求默认值，秒杀接口HTTP请求，聚合报告，CSV Data Set Config（CSV数据文件设置）,创建过程如下图：

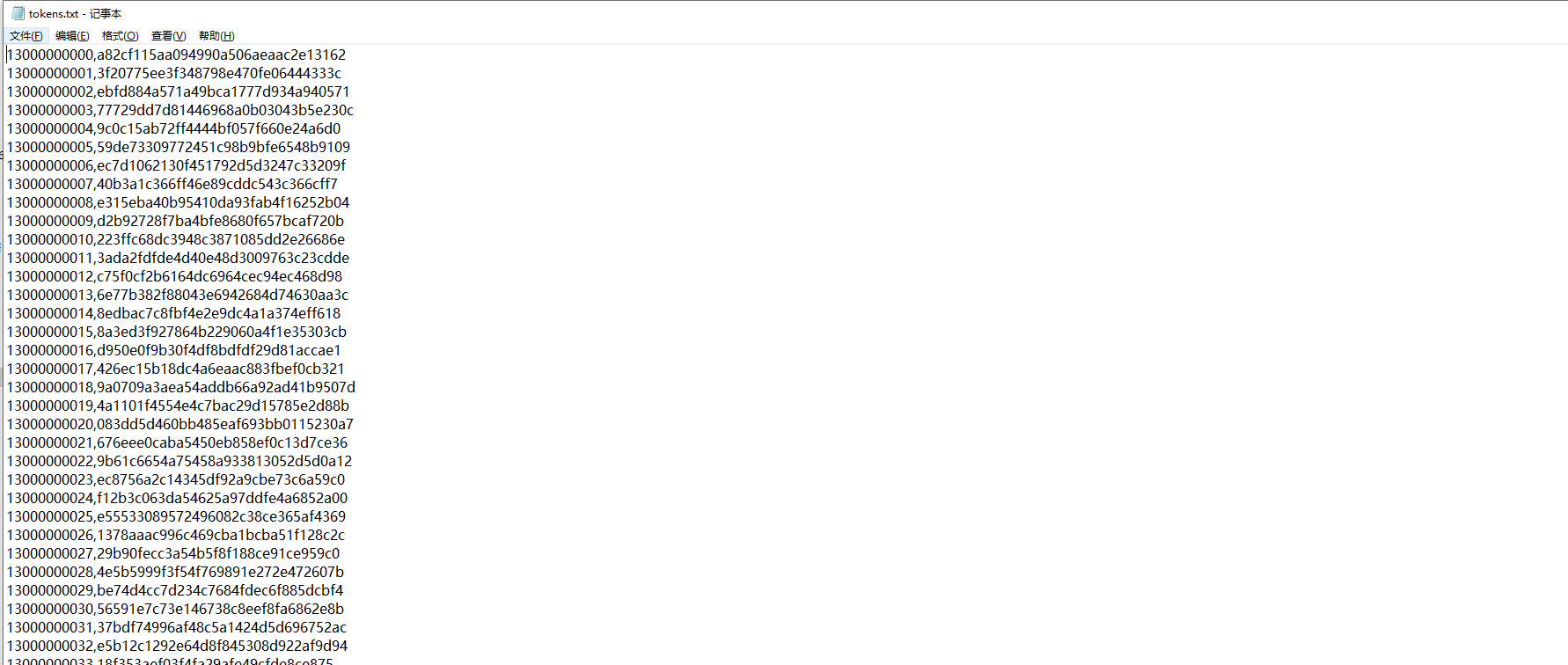








tokens.txt文件内容（userId,token）如下：



测试接口内容如下：



## Redis压测工具redis-benchmark

### 一、redis压测常用指令

1.redis-benchmark -h 127.0.0.1 –p 6379 -c 100 -n 100000 （-c:并发数或用户数 -n:请求数）

命令行解释：100个并发连接，100000个请求

2.redis-benchmark -h 127.0.0.1 -p 6379 -q -d 100 （-q:查询操作 -d:字节数）

命令行解释：存取大小为100字节的数据包

3.redis-benchmark -t set,lpush -q -n 100000

命令行解释：只测试set和lpush命令(只测试某些操作的性能)

4.redis-benchmark -n 100000 -q script load “redis.call(‘set’,’foo’,’bar’)”

命令行解释：只测试某些数值存取的性能（使用直接命令来运行）

注意：Redis能通过Pipelining管道来对tps吞吐量进行优化

参考博客：

<https://blog.csdn.net/nihaomanihao11/article/details/73824544>（Redis自带的redis-benchmark使用）

<https://blog.csdn.net/qq_22152261/article/details/79434206>（Redis的管道操作（Pipelining）评测）