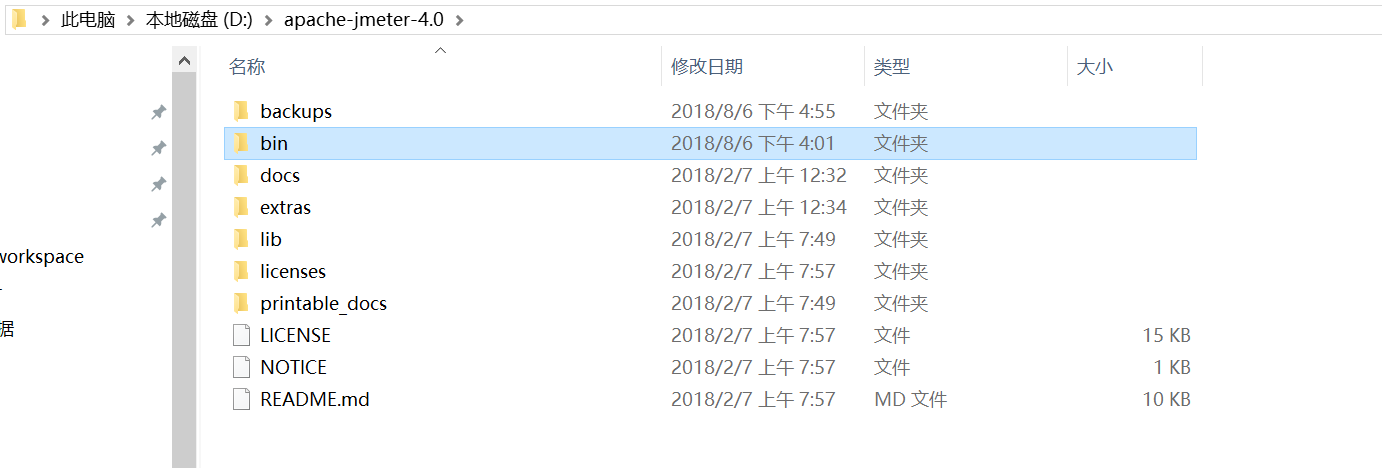
# Windows

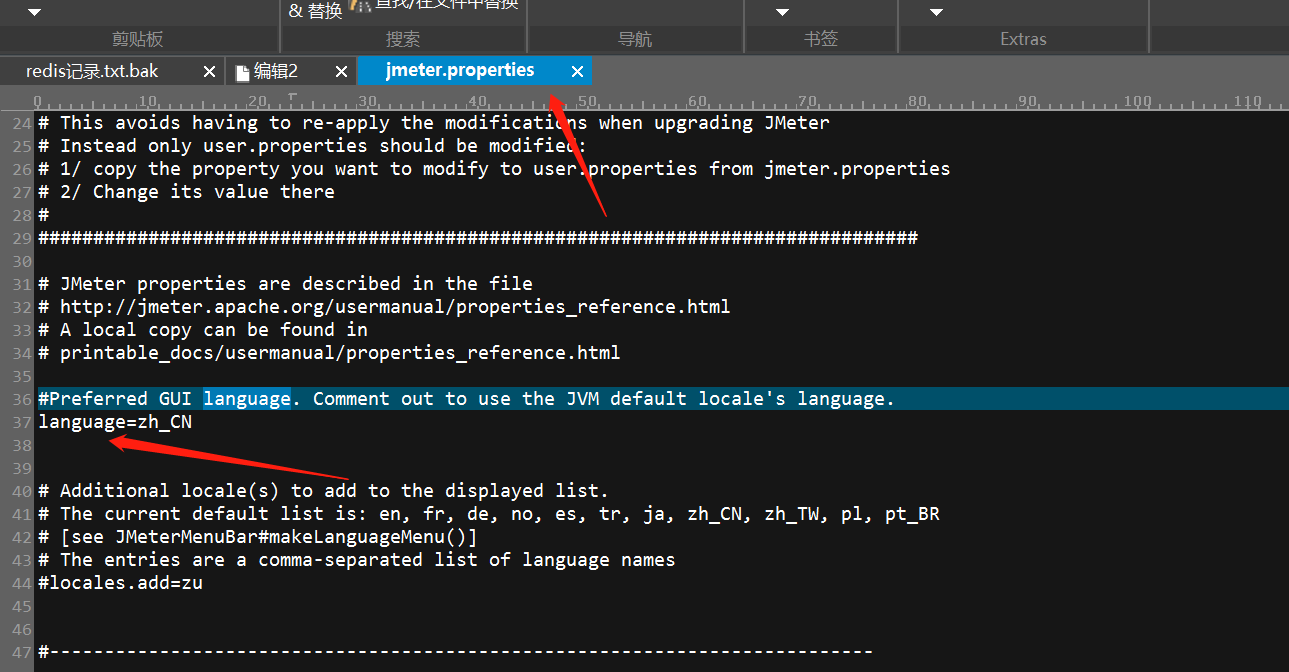
## 安装

apache-jmeter-4.0.zip解压后可直接使用（JDK1.8）

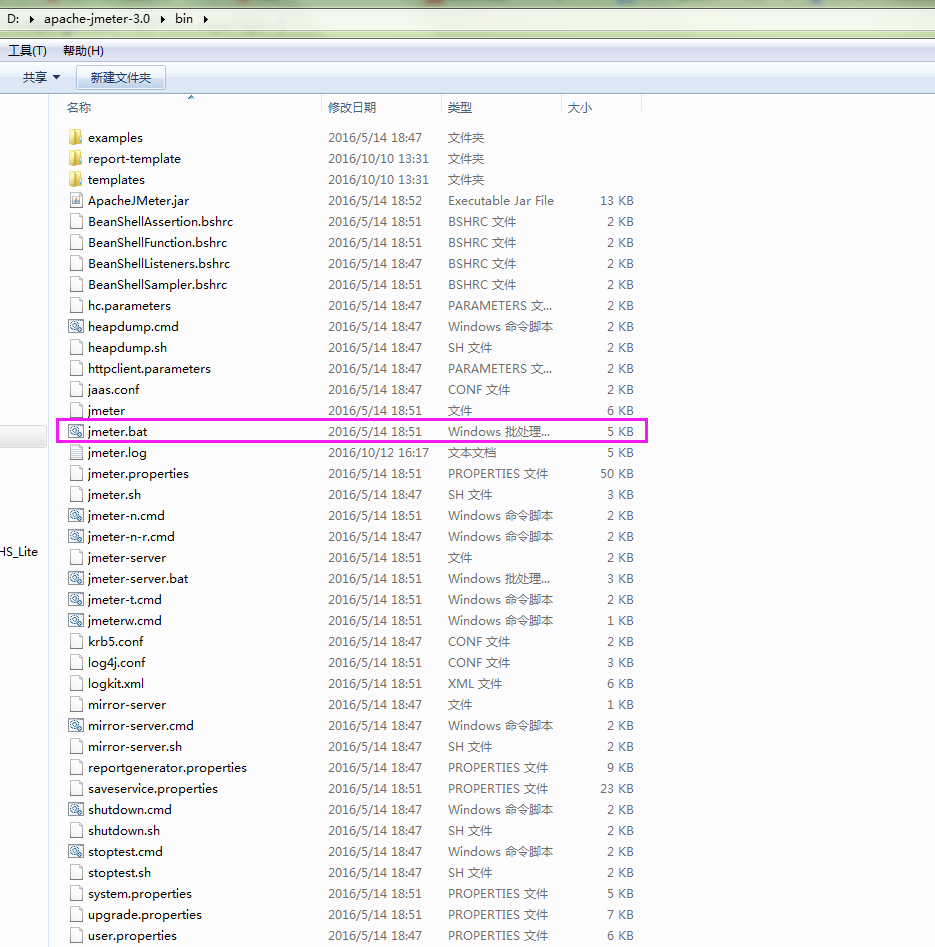
下载后无需安装，解压后即可使用。下图为解压后的目录。



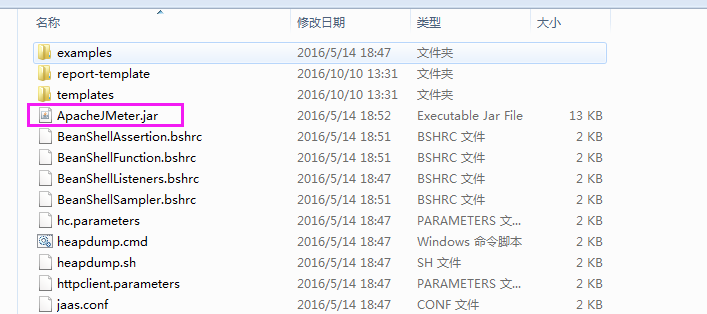
切换成中文界面切换成中文界面



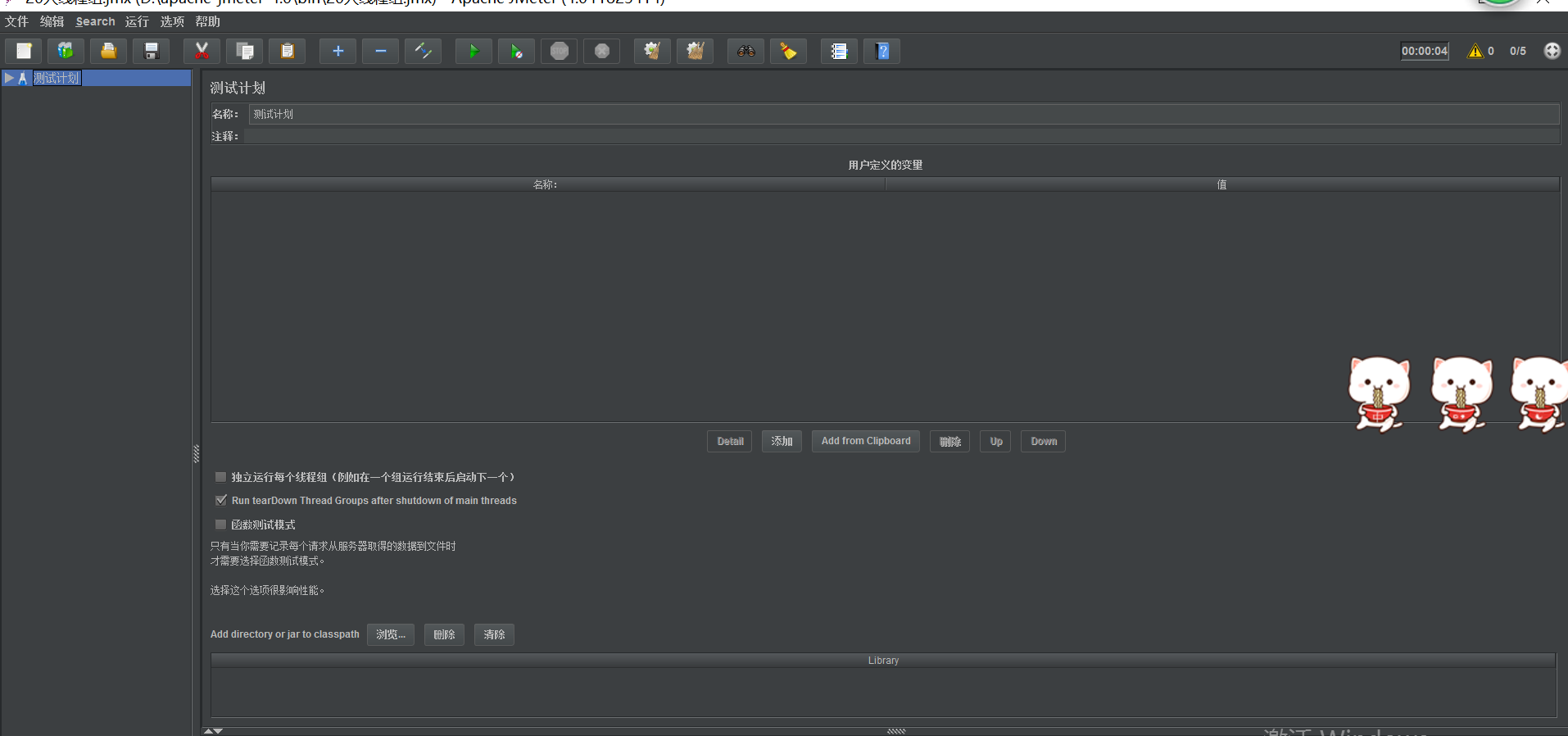
操作完上面的步骤，我们就可以点击Jmeter中bin目录下面的jmeter.bat文件即可打开Jmeter了。  
注意：打开的时候会有两个窗口，Jmeter的命令窗口和Jmeter的图形操作界面，不可以关闭命令窗口。



 备注：也可以通过ApacheJMeter.jar打开。



进入JMeter界面。



## Jmeter核心组件介绍

### 测试计划（Test plan）？

    就是压测的一个流程，比如压测什么接口，并发多少用户，协议等等，在测试计划下面配置线程组，采样器，监听器等组件

什么是线程组(Thread Group)？

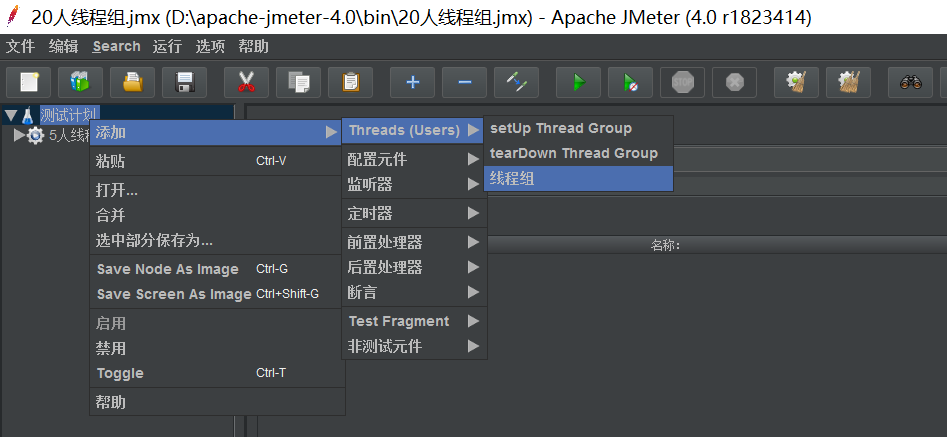
线程组就是模拟多少用户并发访问你的应用，这个组里面共有多少用户，创建方法如上图现在“添加->threads->线程组“

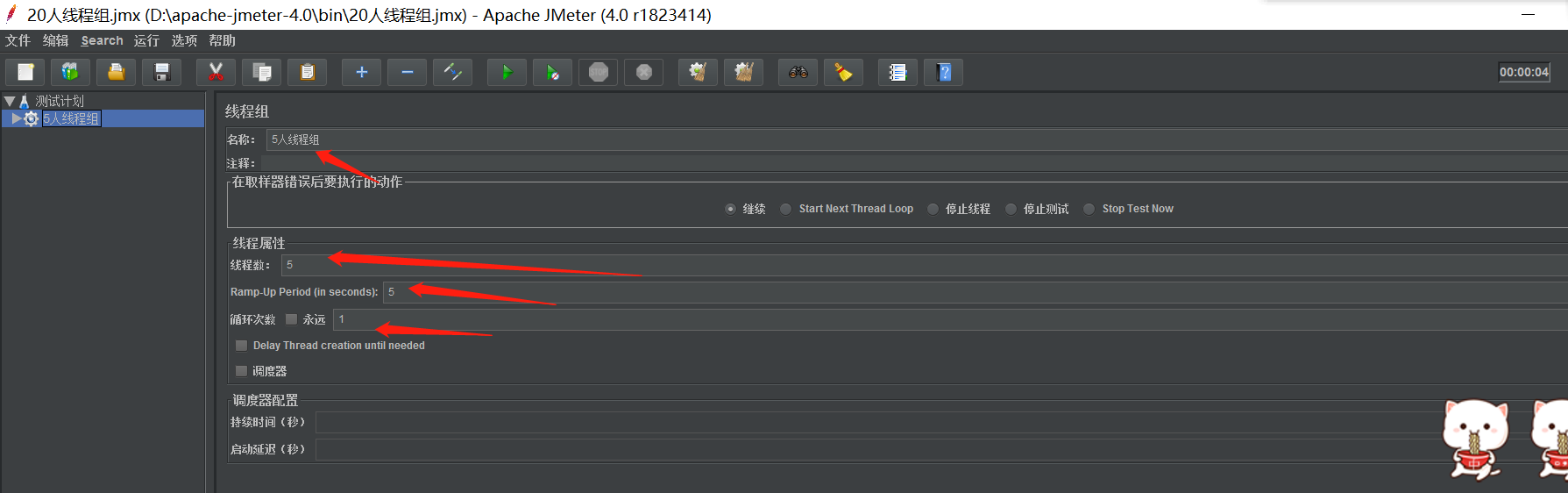
    参数说明：

    线程数：虚拟用户数。一个虚拟用户占用一个进程或线程

    准备时长（Ramp-Up Period(in seconds)）：全部线程启动的时长，比如100个线程，20秒，则表示20秒内100个线程都要启动完成，每秒启动5个线程

    循环次数：每个线程发送的次数，假如值为5，100个线程，则会发送500次请求，可以勾选永远循环





### 采样器(Sampler)

真正干活的组件，采样器里面配置你要压测的接口，协议，参数等

可以创建各种采样器，如Http请求，FTP请求，JDBC压测数据库请求，JMS等，对应的采样器有不同的配置

以http请求采样器为例

名称：采样器名称，一般是接口名称，方便后续进行区分，比如“login接口”

注释：采样器的描述，方便别人理解，类似压测接口描述

web服务器配置

协议：压测的协议，比如“http”或者"https"

服务器名称或IP：压测目标机器的端口或者服务器IP地址，比如 www.xdclass.net

端口号：压测目标机器的端口号，比如 8080

HTTP请求配置

方法：http的请求方法，常用的有GET,POST,PUT,DELETE,PATCH等

路径：压测的接口的URL，比如 /api/v1/users

Content encoding:是否进行内容编码

自动重定向：http有302状态码返回的时候，重定向，但不会产生记录，不可以做关联请求,

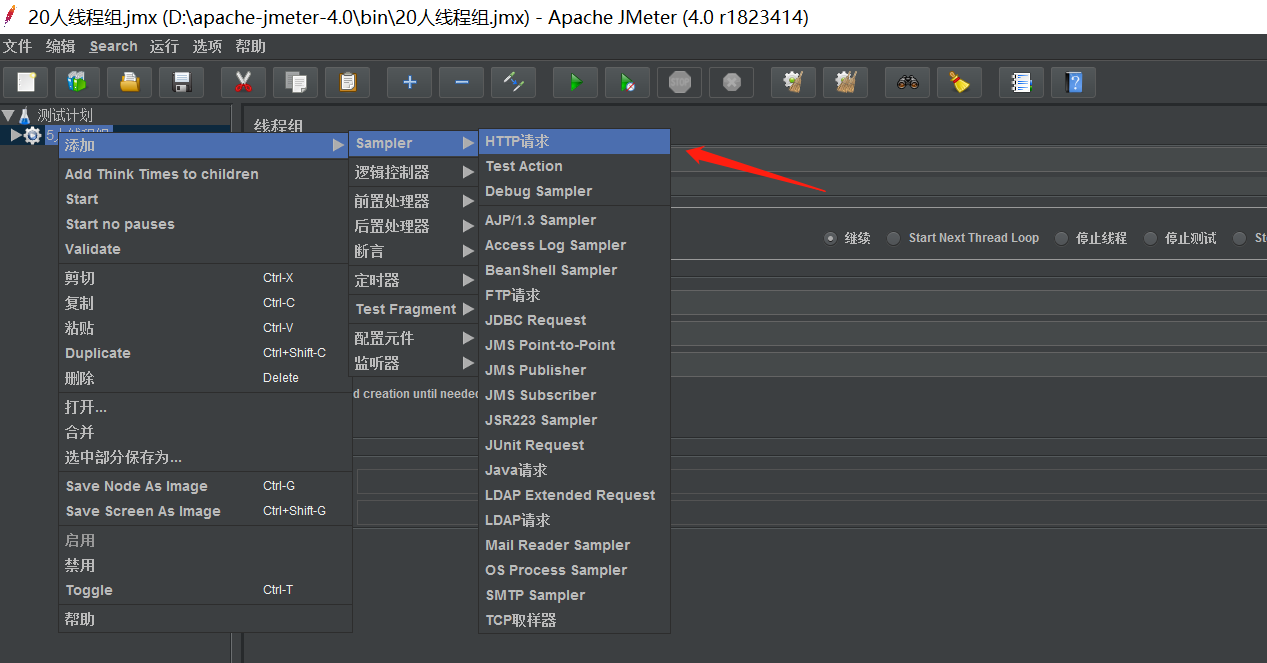
比如比如第二个请求要用到第一个请求的数据，则选择自动重定向无法做关联

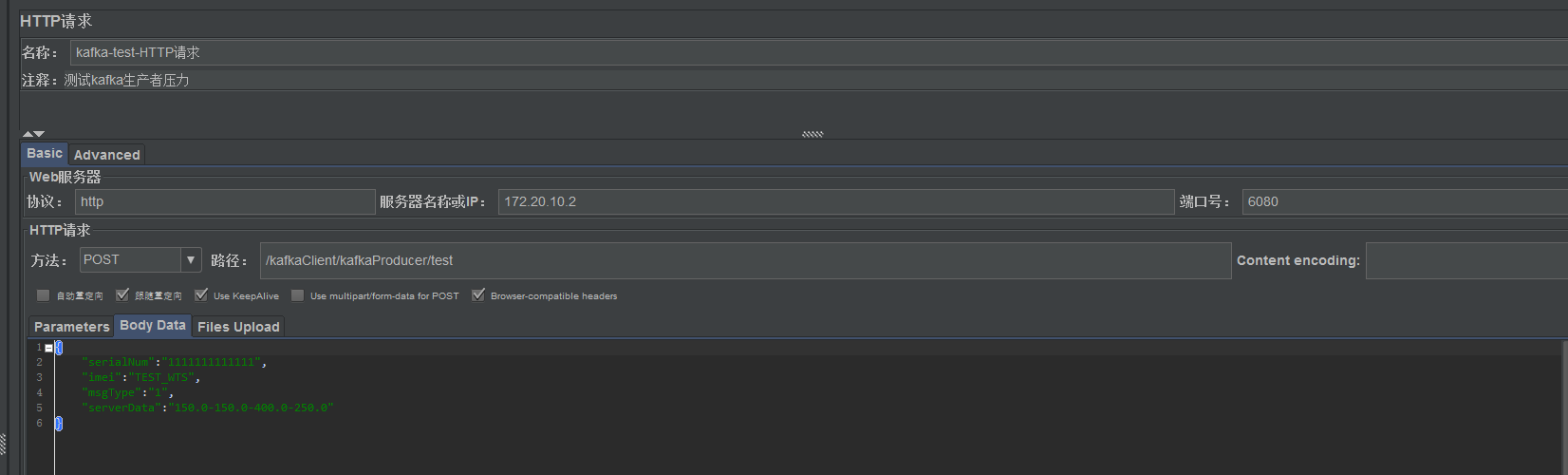
跟随重定向：会产生记录，可以做关联，默认勾选这个

Use multipart/from-data for HTTP POST ：当发送POST请求时，使用Use multipart/from-data方法发送，默认不选中。

parameters：参数配置

Files Upload：文件上传配置





### 结果树

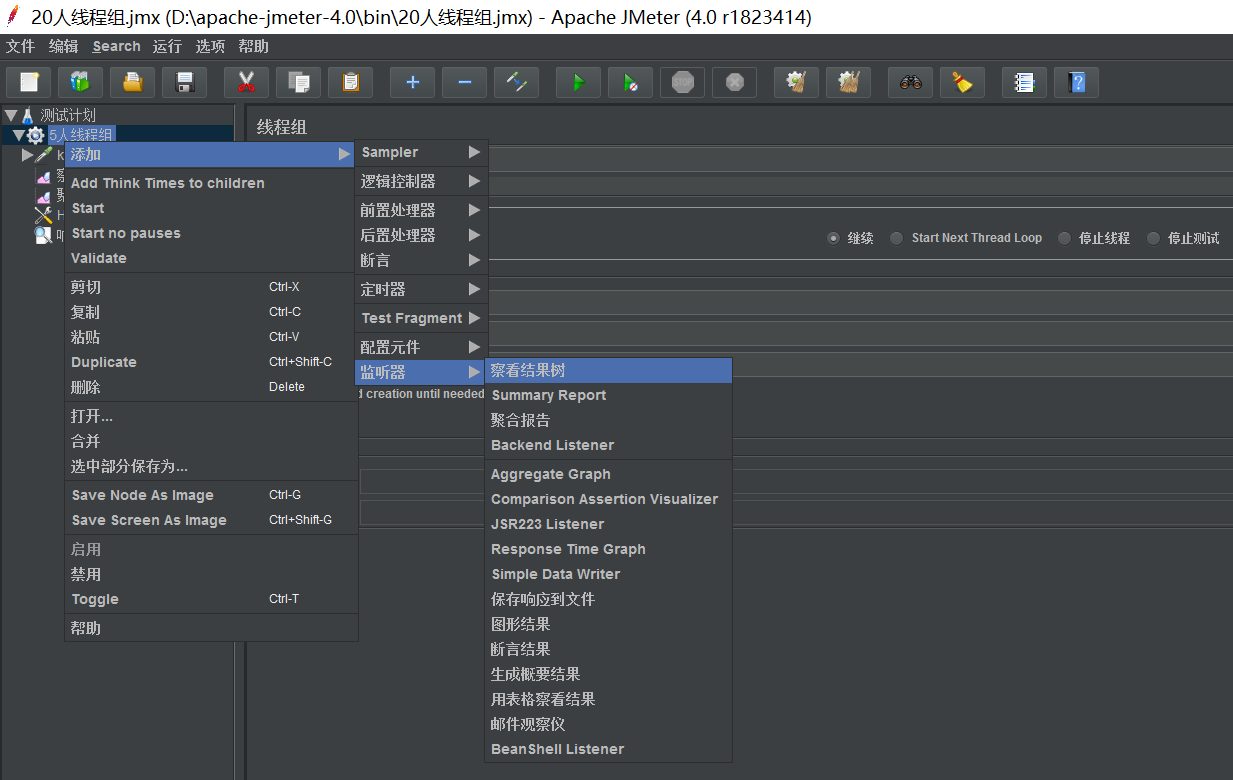
压测的结果查看，可以看到压测发送的url，http参数，返回结果，返回的整个响应，然后如果有断言或者调试，可以看到断言是否通过等等

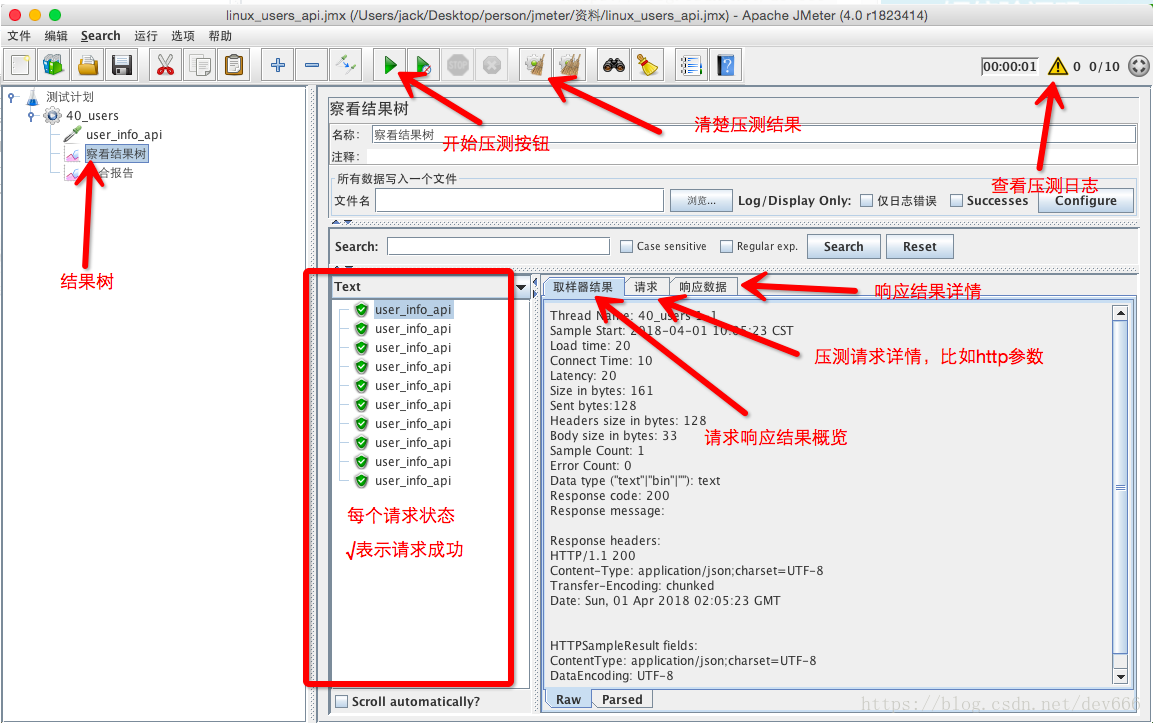
添加路径： 线程组->添加->监听器->察看结果树

如果不添加结果数，则点击压测开始按钮，压测具体响应我们无法知道，所以需要添加结果数，

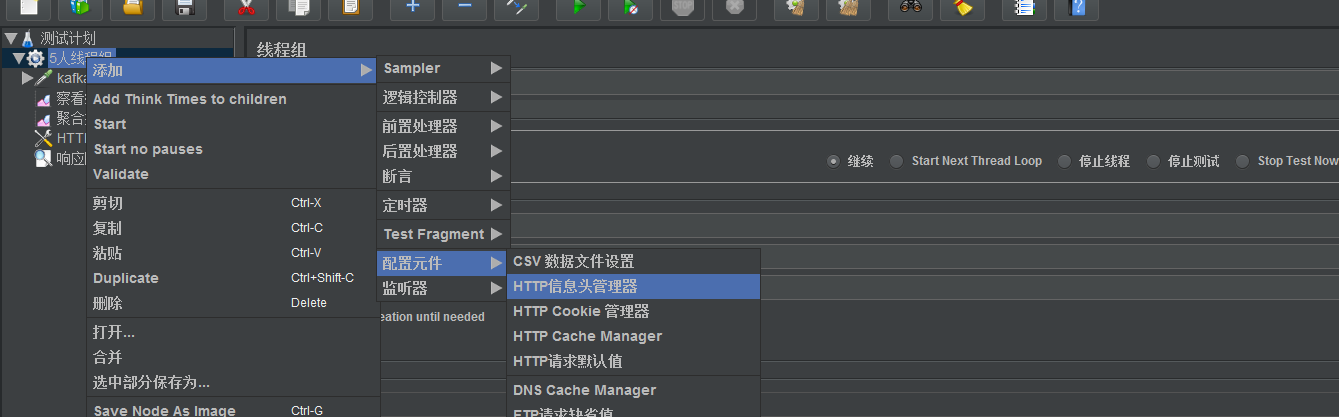
一来是方便调试压测脚本，看请求参数，路径是否正常

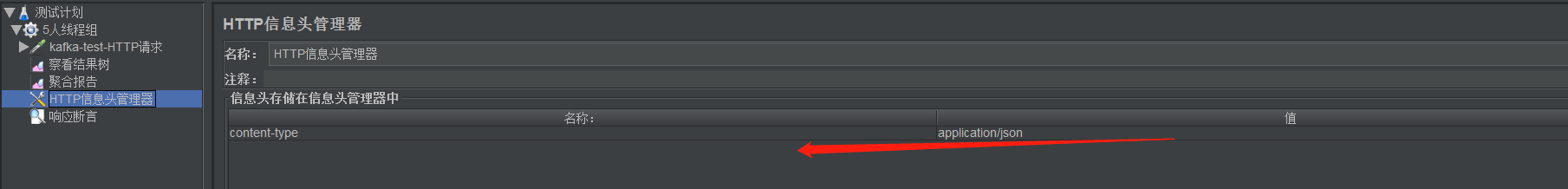
切换到结果数查看页面，点击压测按钮就可以进压测，边压测可以边看到请求的结果。



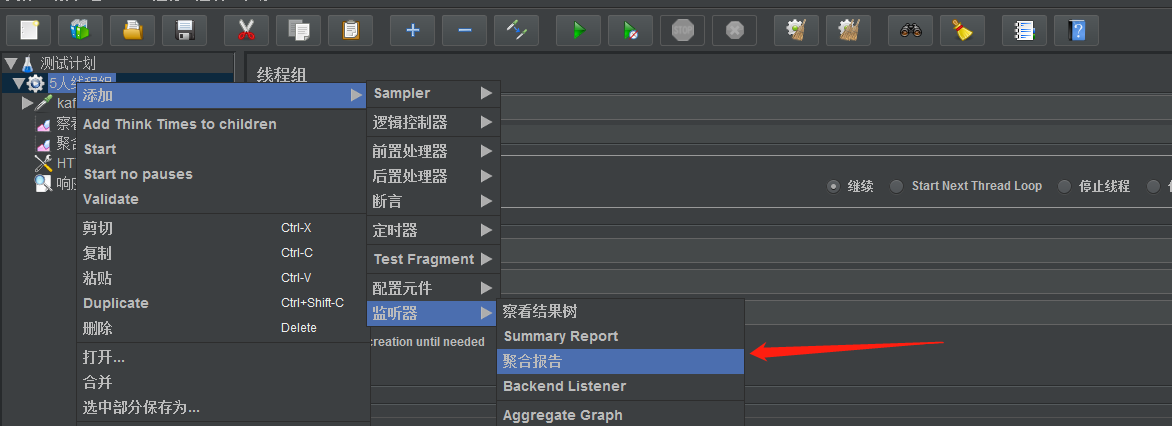


### HTTP信息投管理器



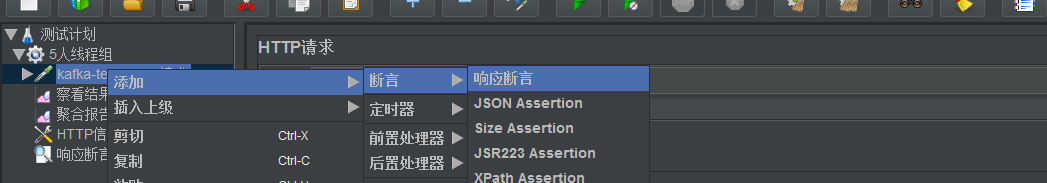


### 聚合测试报告



### 响应断言

如果不添加正确的相应断言，会影响结果树的结果



## 测试

如上所述，测试/kafkaClient/kafkaProducer/test接口

参数为：

{

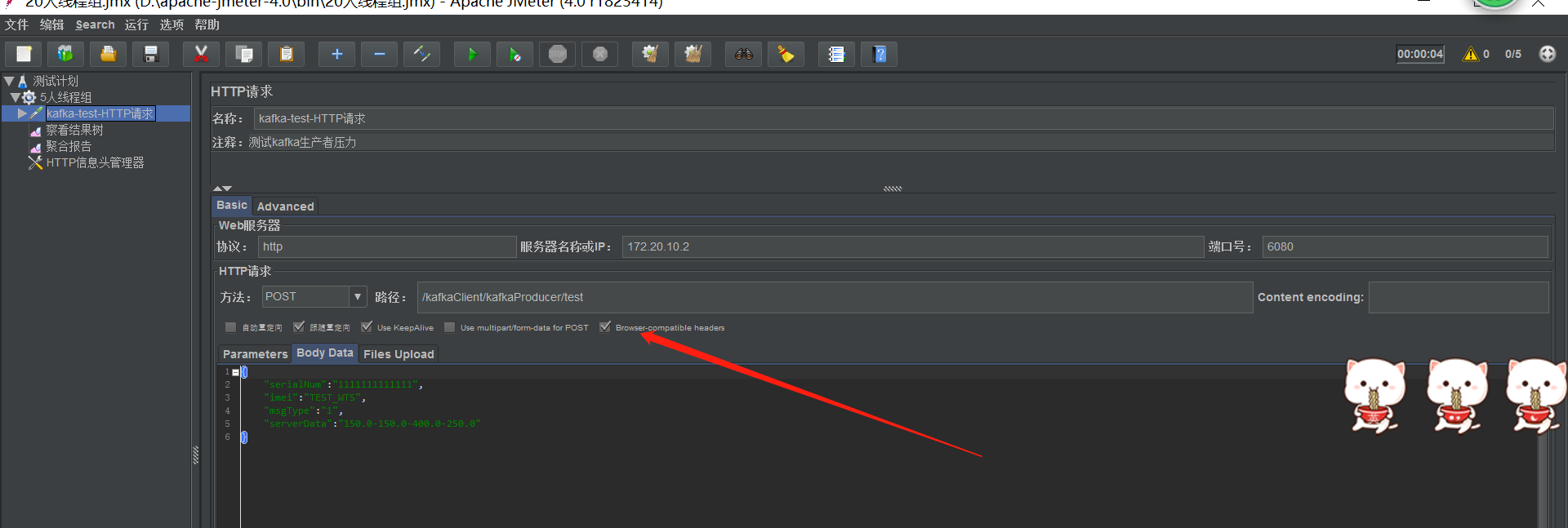
"serialNum":"1111111111111",

"imei":"TEST\_WTS",

"msgType":"1",

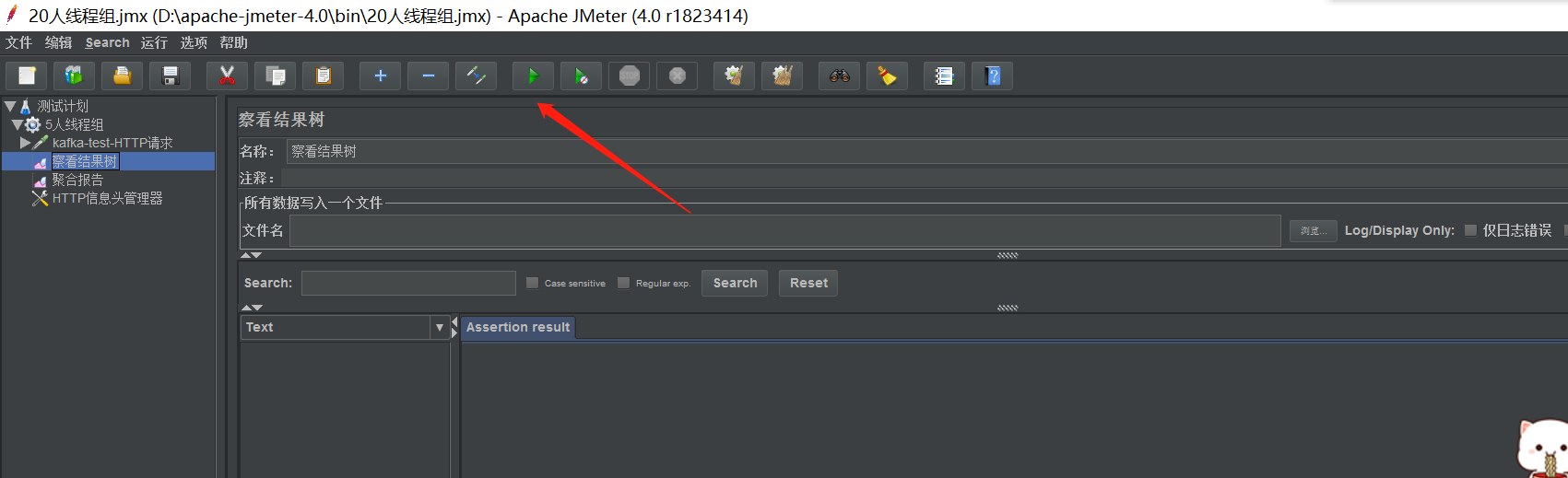
"serverData":"150.0-150.0-400.0-250.0"

}

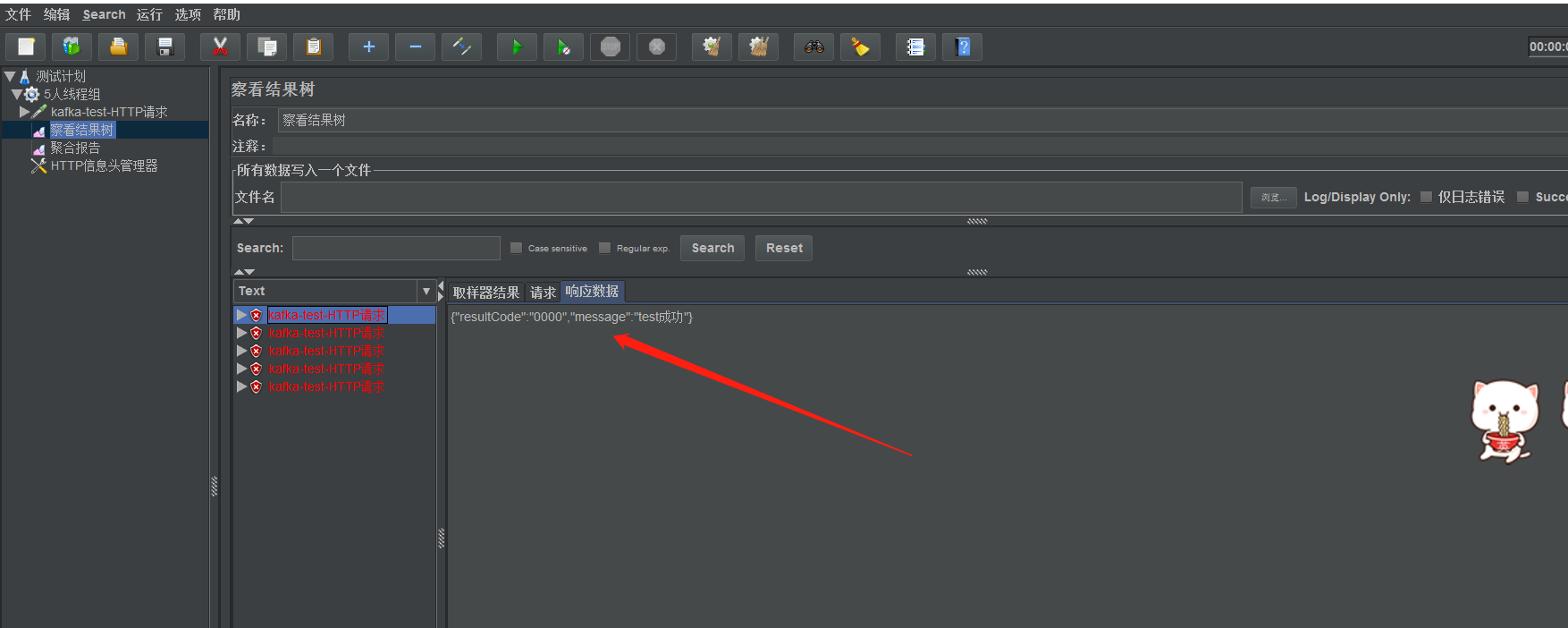


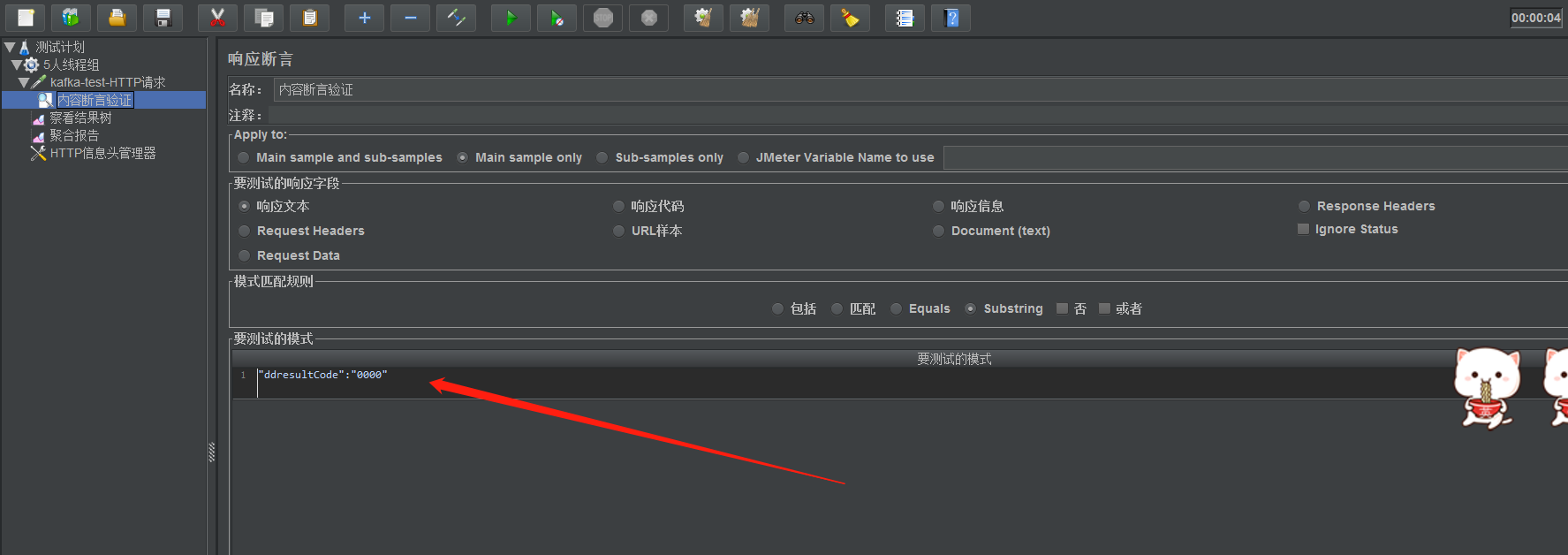
### 执行测试按钮

在查看结果树界面点击执行按钮

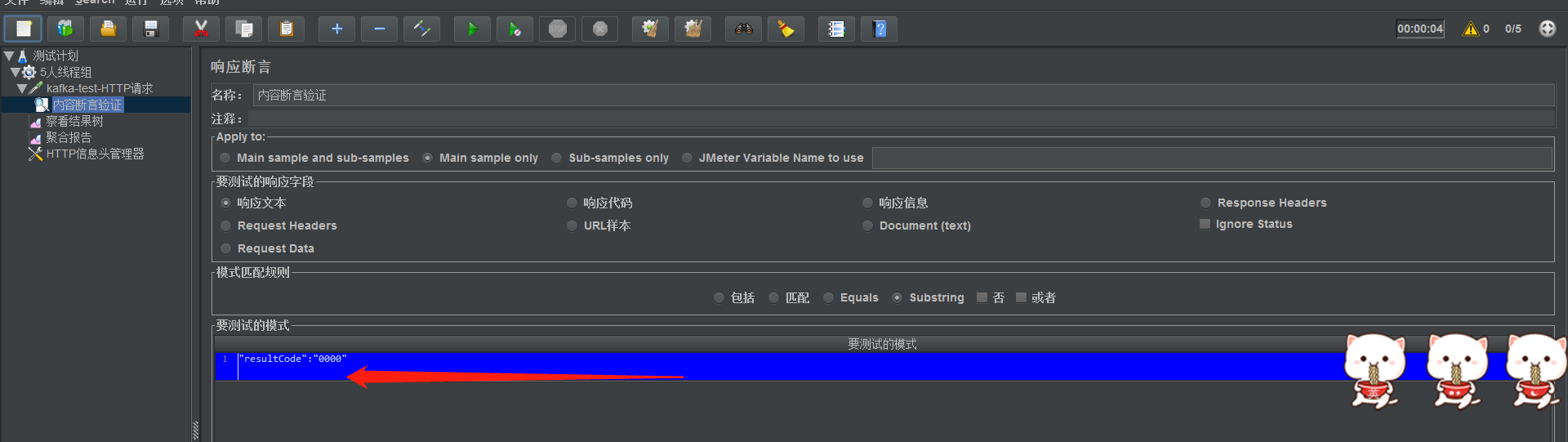


报错，查看相应数据没有问题，查看相应数据和相应断言是否匹配（我这里不匹配，改成匹配的就没有问题了）

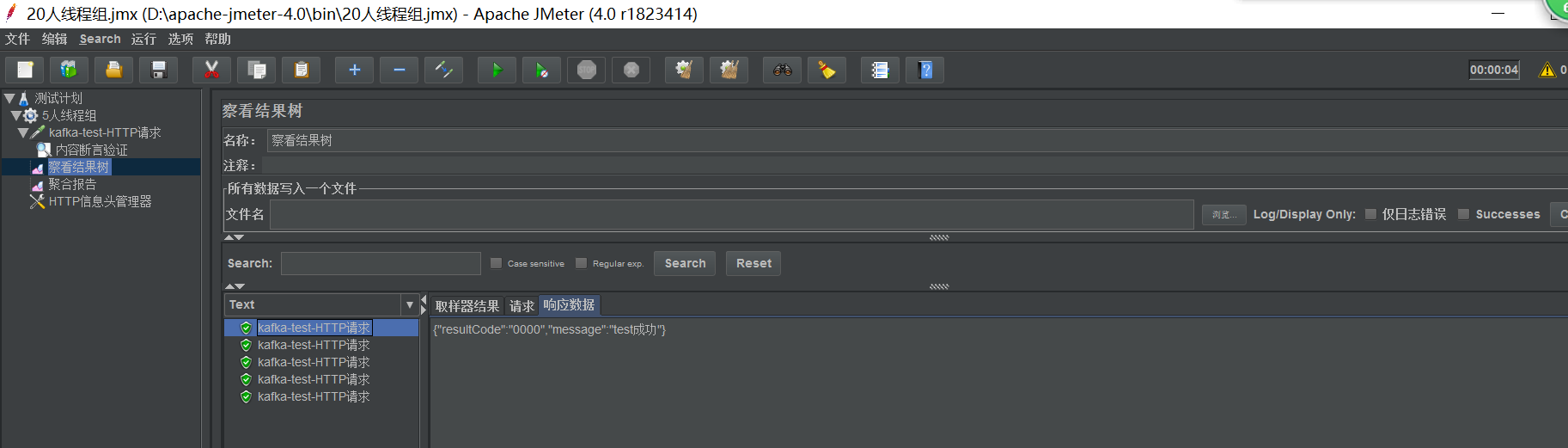




修改相应断言后：



### 查看结果树相应数据

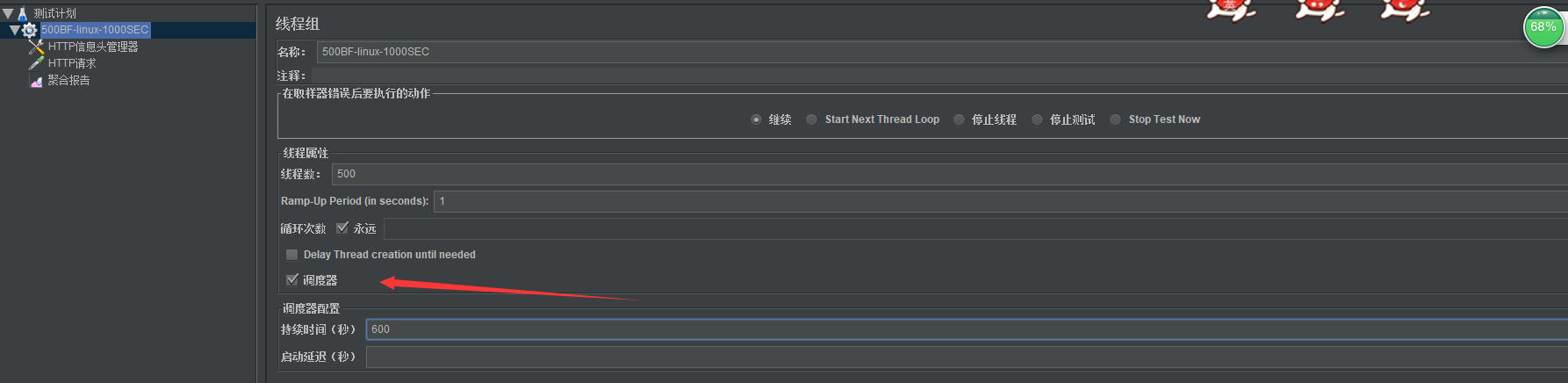


### 查看聚合报告

### 测试类型

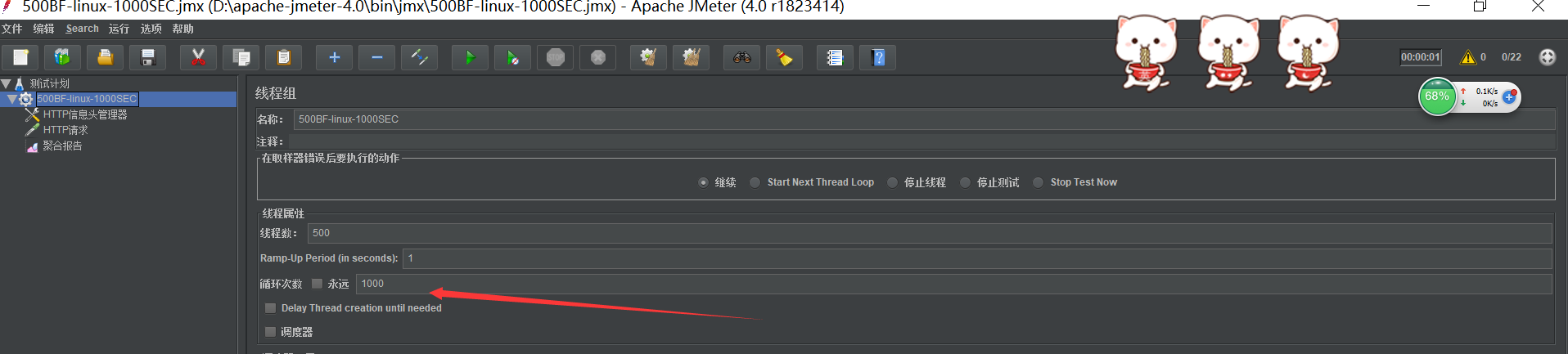
#### 定时测试

线程组勾选调度器，设置持续时间



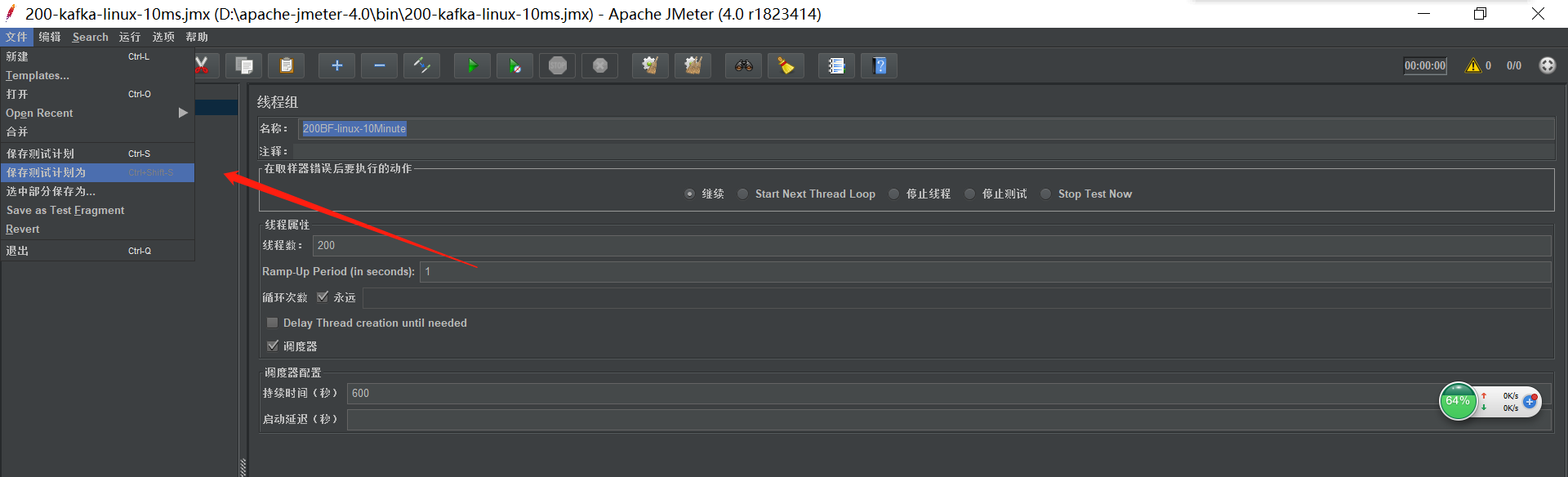
#### 定量测试

固定循环测试



## 保存

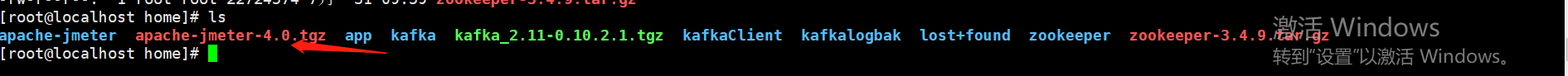
文件，保存测试计划为……



# Linux

## 安装

将下载好的apache-jmeter-4.0.tgz上传到linux指定目录下（10.100.8.216:/home/s）

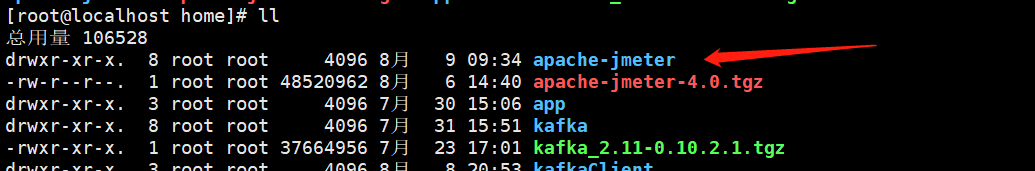


解压（解压后可直接使用）

tar –zxvf apache-jmeter-4.0.tgz

重命名

mv apache-jmeter-4.0.tgz apache-jmeter



然后进入bin目录查看是否安装成功

cd /home/apache-jmeter/bin/

sh jmeter.sh –v

看到如下结果说明安装成功



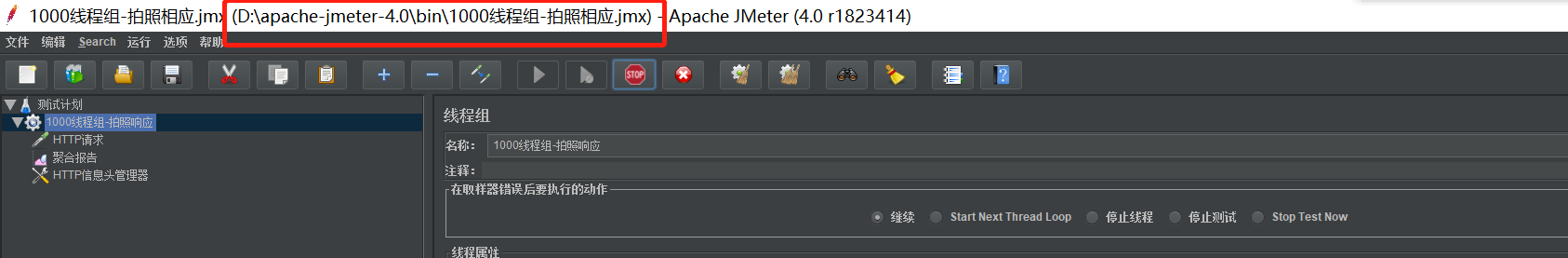
## 测试

### 步骤一

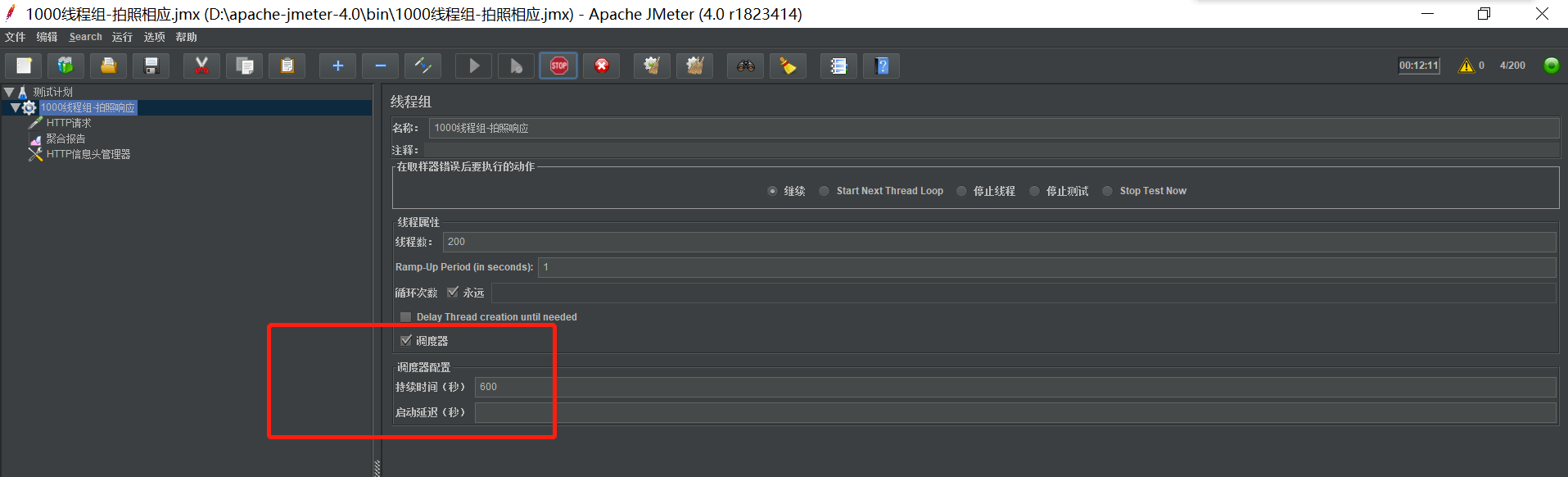
在win上的Jmeter中编写好脚本，并将脚本保存为.jmx文件（脚本可不添加监听器）

**说明：**

在windows中界面操作测试保存后，会生成.jmx文件（不要出现中文），找到相应的文件即为脚本，由于我要将此脚本放到linux下测试，故设置了测试时间为十分钟，ip和端口改为内网如下图所示



调度器配置



### 步骤二

在Linux上进到解压apache-jmeter-3.2的路径下，执行Jmeter脚本的命令是：

**./bin/jmeter -n -t .jmx文件（脚本） -l .jtl文件（测试运行结果文件）**

例：

./bin/jmeter -n -t 200-kafka-linux-10ms.jmx -l 200-kafka-linux-10ms.jtl

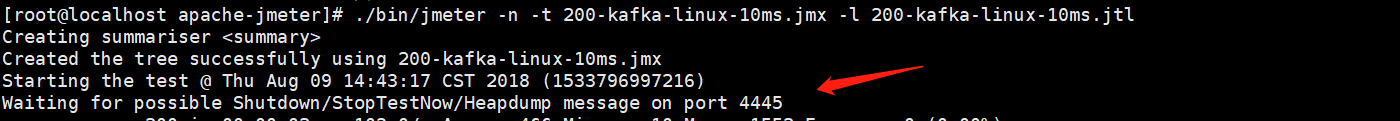
**参数说明：**

-n表示以nogui方式运行测试计划

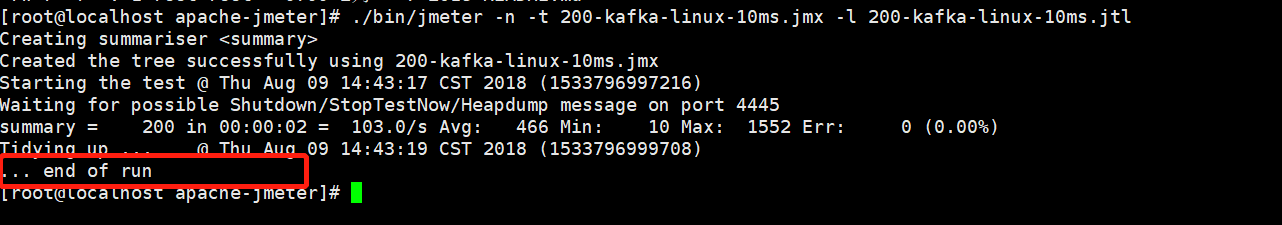
-t表示测试计划，后面跟测试计划名称

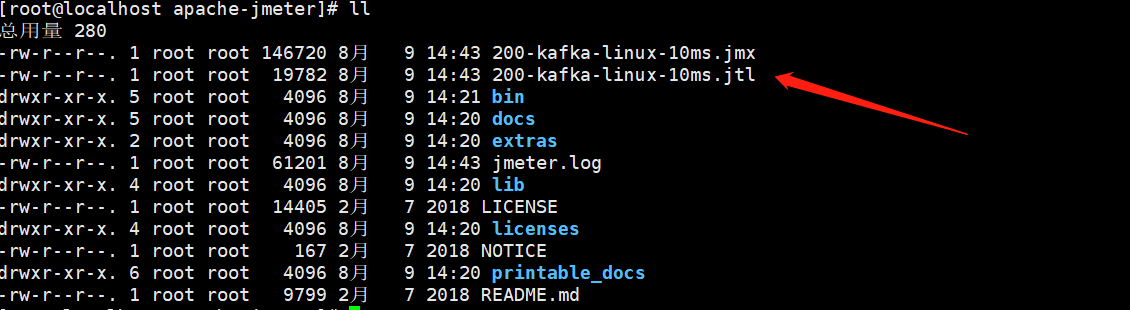
-l表示测试结果，后面跟测试结果文件名称

**运行后显示如下界面，即成功运行了脚本：**



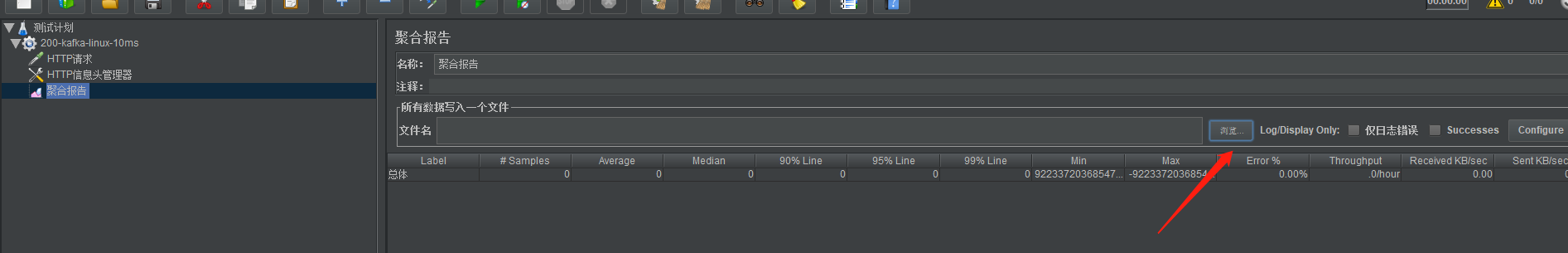
**运行结束后会显示**

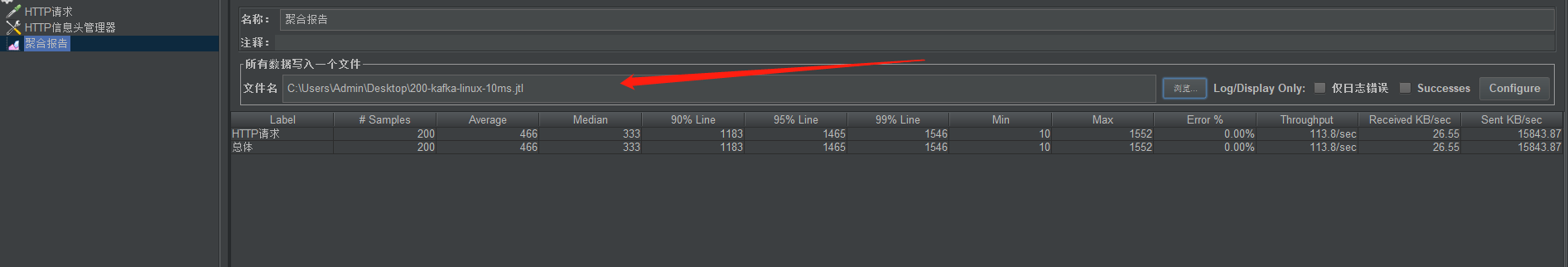
显示 ... end of run 即脚本运行结束，在输入的测试运行结果文件的路径下就会出现命令中输入的jtl文件了。



3、将上一步运行结束后生成的.jtl文件拷贝到win上，打开win上的Jmeter，随便新建一个项目，也可以用之前的项目，添加监听器，在监听器界面点击浏览，选择该.jtl文件，就可以查看结果了。

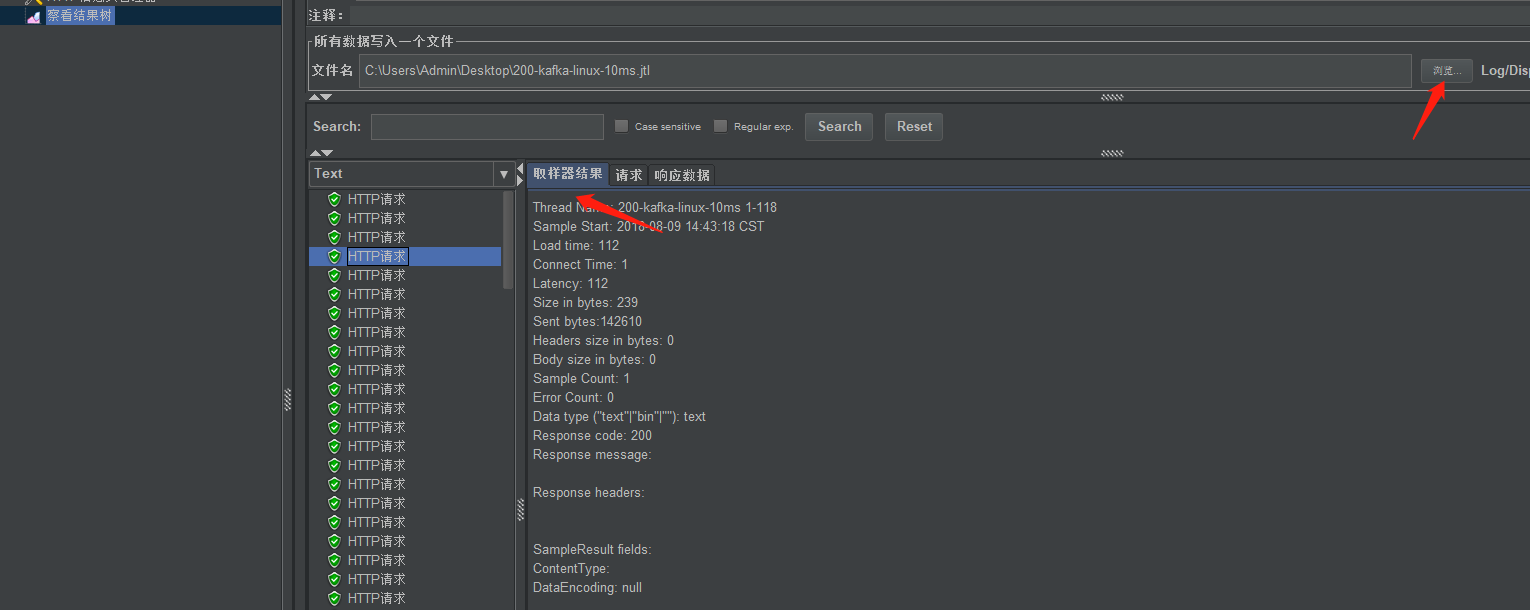
查看聚合报告





查看结果树

打开文件后，就会显示文件中的结果，但是如上述给出的命令在linux下得到的jtl文件里只有取样器结果，请求和相应数据为空



备注： 另外，在Linux下我们有时候希望线程可以在后台运行，这样我们关闭当前连接后，线程依然可以运行，这里提供一个将 jmeter命令设置为后台线程的方法。

使用setsid命令：

 setsid bin/jmeter -n -t .jmx文件 -l .jtl文件

 setsid ./bin/jmeter -n -t .jmx文件 -l .jtl文件

有没有 ./ 当前目录的表示符都可以

## 注意

1. IP和端口改为内网204IP和80端口，
2. 去掉代理，否则会报错
3. 保存.jmx文件时注意保存整个测试计划

# Kafka客户端安装位置

## 216（两台）

/home/kafkaClient

/home/kafkaClient6081

## 217（两台）

/home/kafkaClient

/home/kafkaClient6081

## 218（一台）

/home/kafkaClient