**实验九 性能测试**

1. **实验目的**

（1）了解Jmeter性能测试工具的用途和简单的操作；

（2）掌握Jmeter性能测试工具测试过程；

（3）能够使用Jmeter进行负载测试；

1. **实验环境**

架构Demo系统，Jmeter

1. **实验内容**

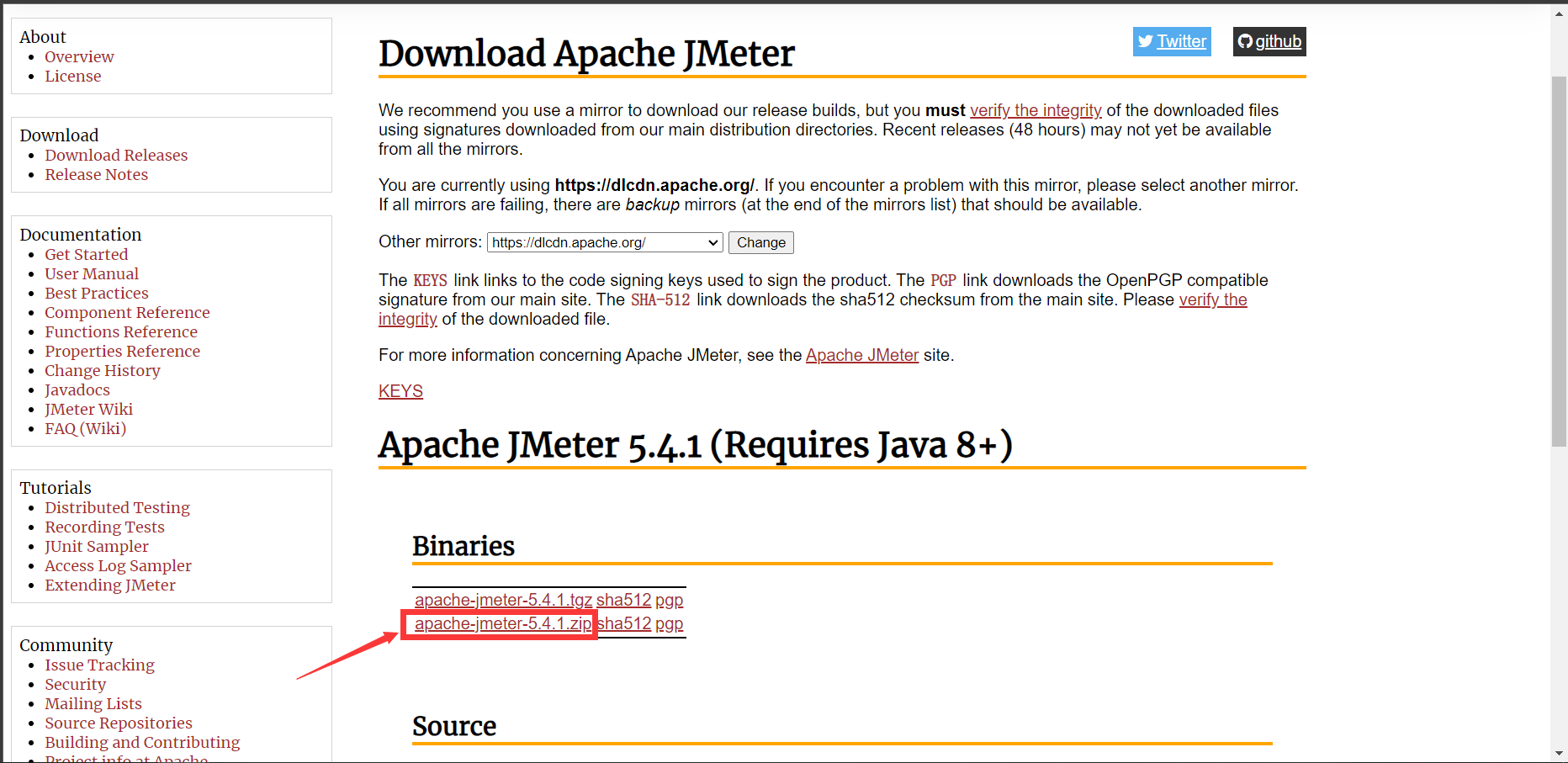
下载并安装Jmeter，在架构Demo系统上进行测试脚本的编写，进行性能测试，并对测试结果进行分析

1. **实验步骤**

4.1 Jmeter的安装和运行

1.安装jdk（需要Java8即Jdk1.8）

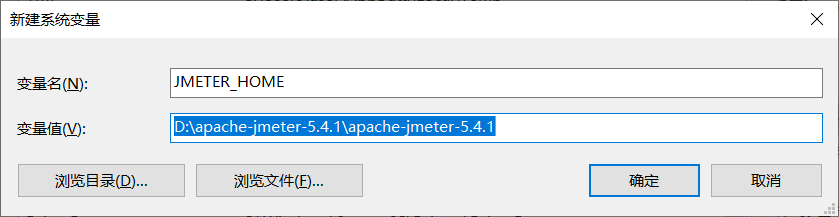
2.前往官网进行下载<https://jmeter.apache.org/download_jmeter.cgi>，并解压压缩包到英文目录下



3.配置环境变量

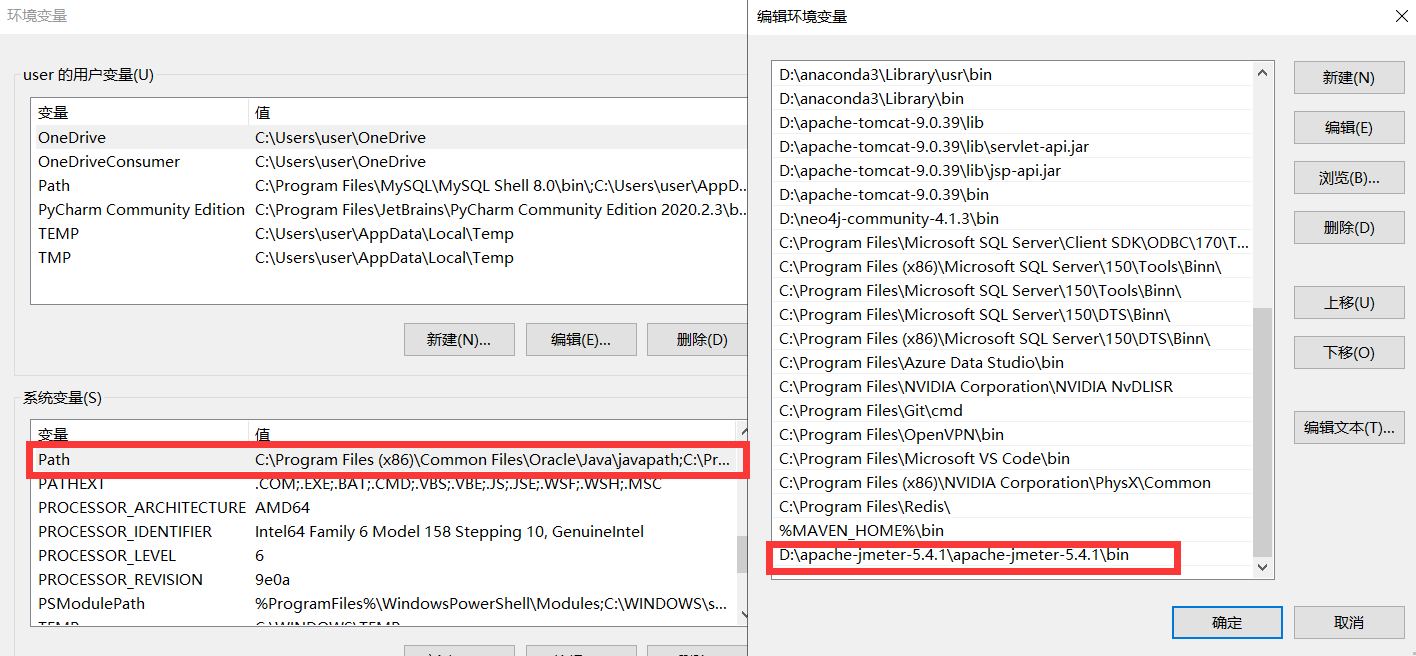
(1) 设置jmeter解压目录的JMETER\_HOME环境变量

系统变量中新建变量，变量名为JMETER\_HOME，变量值为jmeter安装的路径



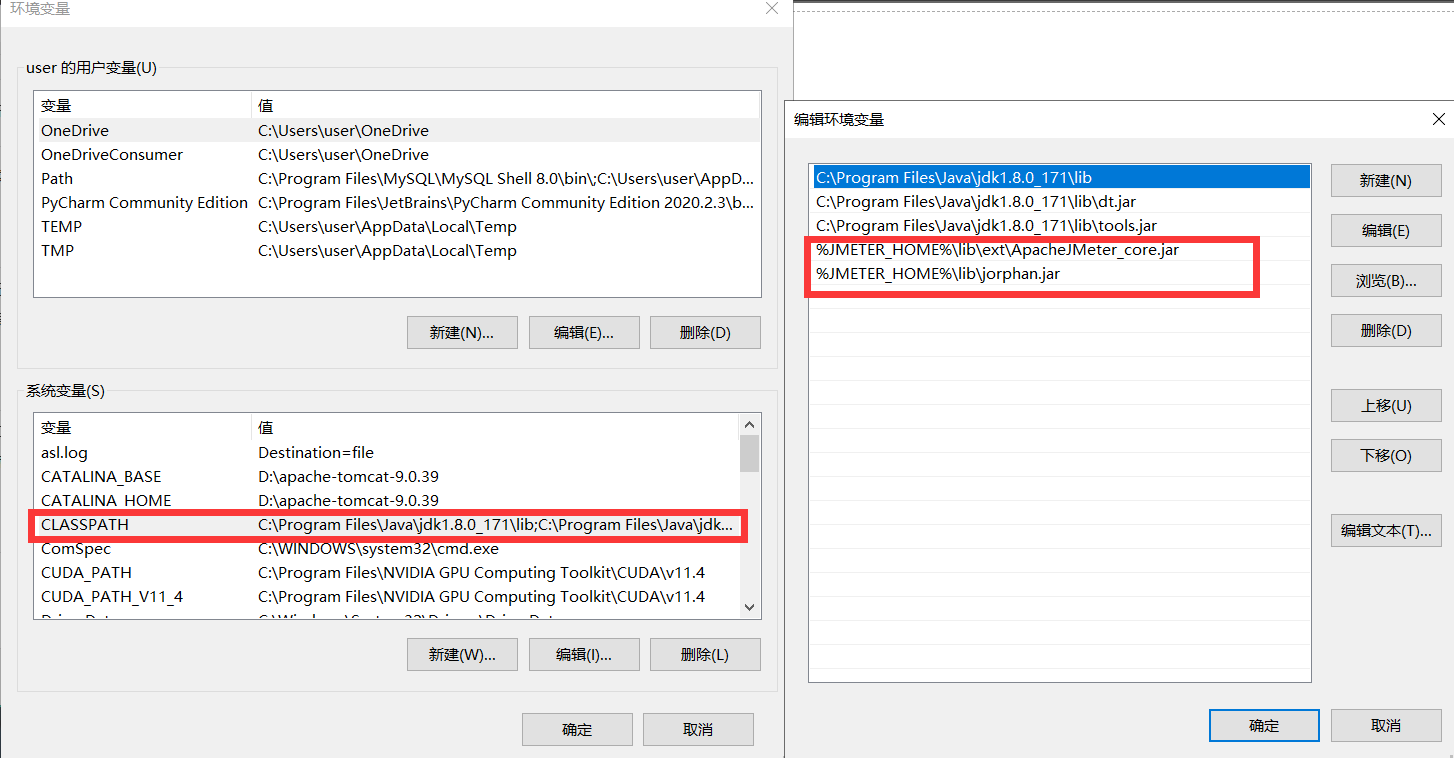
(2) 设置jmeter中bin目录的path环境变量

在系统的PATH变量中添加jmeter安装目录下bin目录的路径

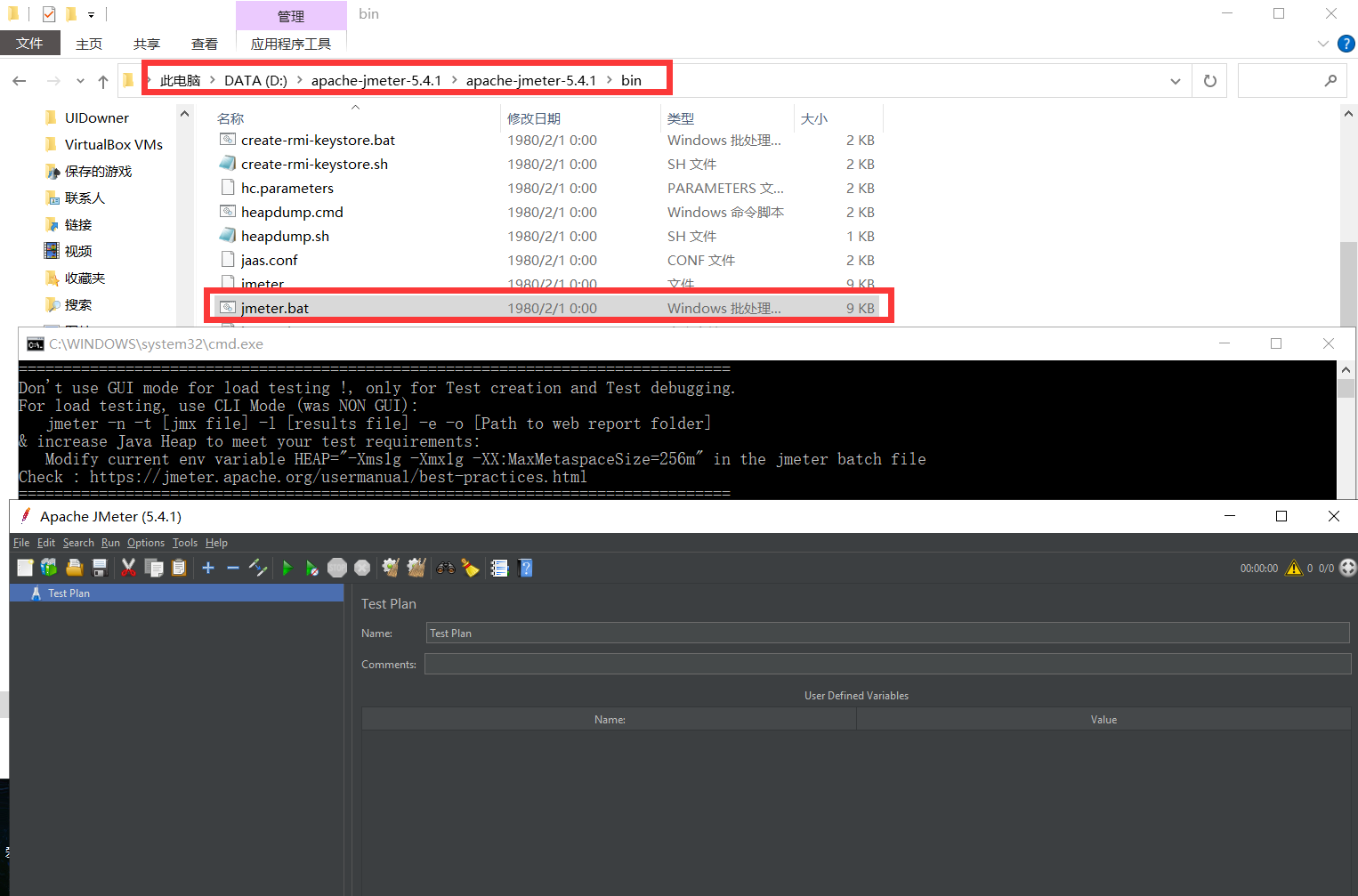


(3) 设置jmeter的classpath变量

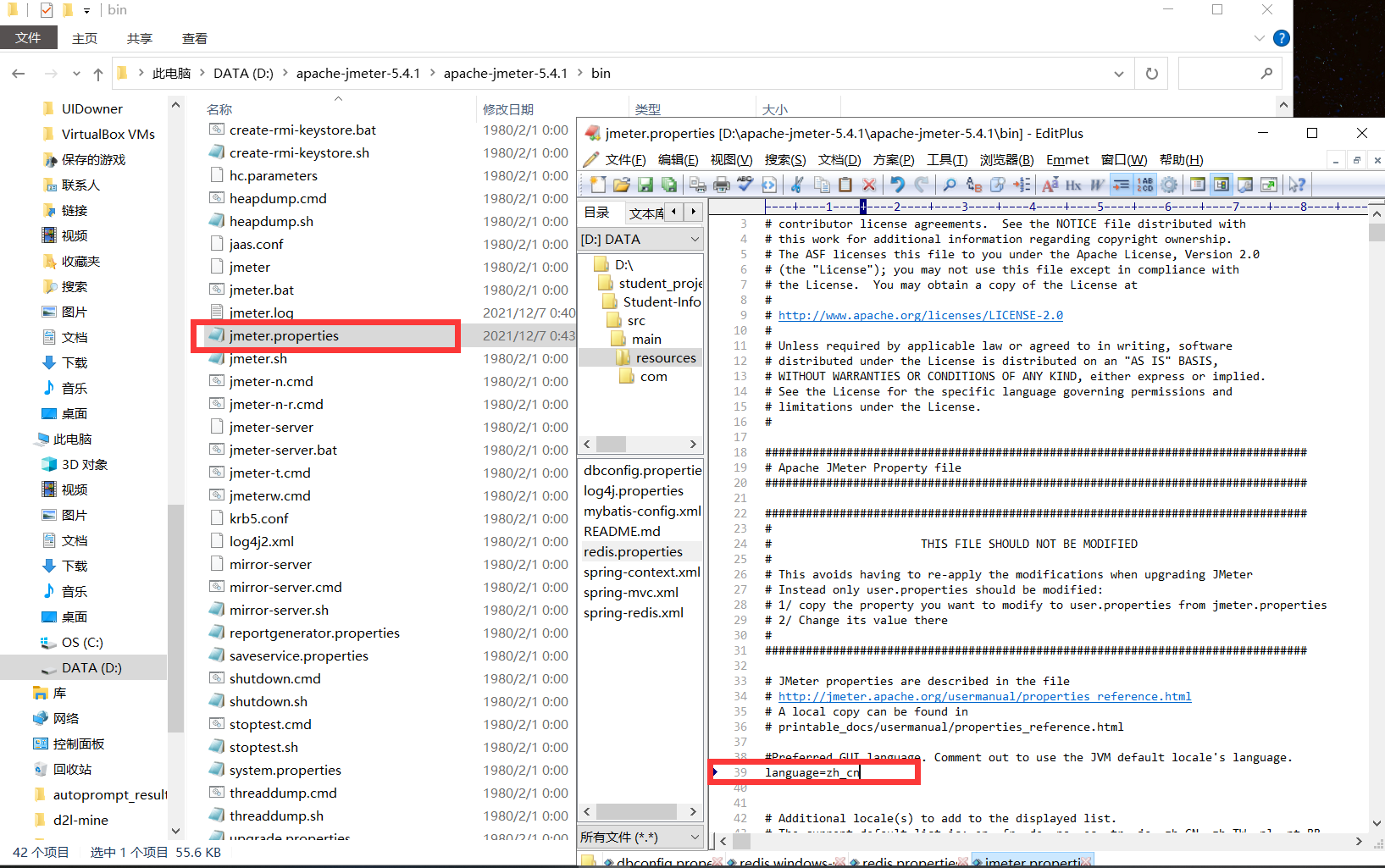
系统属性中找到一个变量名为CLASSPATH的系统变量（若没有则新建），新增两条：%JMETER\_HOME%\lib\ext\ApacheJMeter\_core.jar;%JMETER\_HOME%\lib\jorphan.jar;



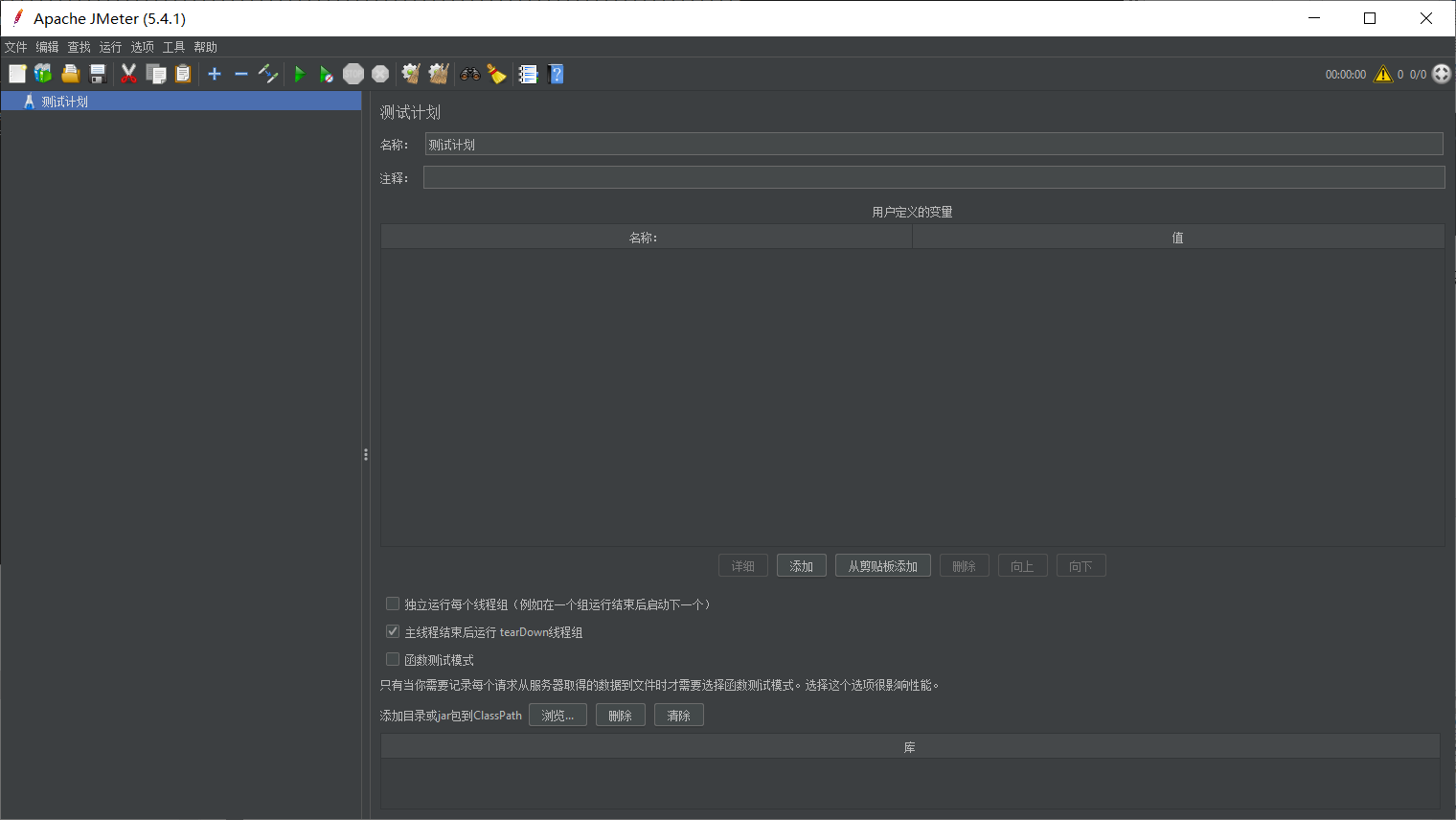
4.进入jmeter下的bin目录，点击打开jmeter.bat文件，即可启动jmeter。



5.由于打开是英文版，所以可以根据需要对软件进行汉化，修改配置文件 jmeter.properties（通过记事本、EditPlus等文件打开），添加或修改以下字段：**language = zh\_cn**，随后重新启动即可。

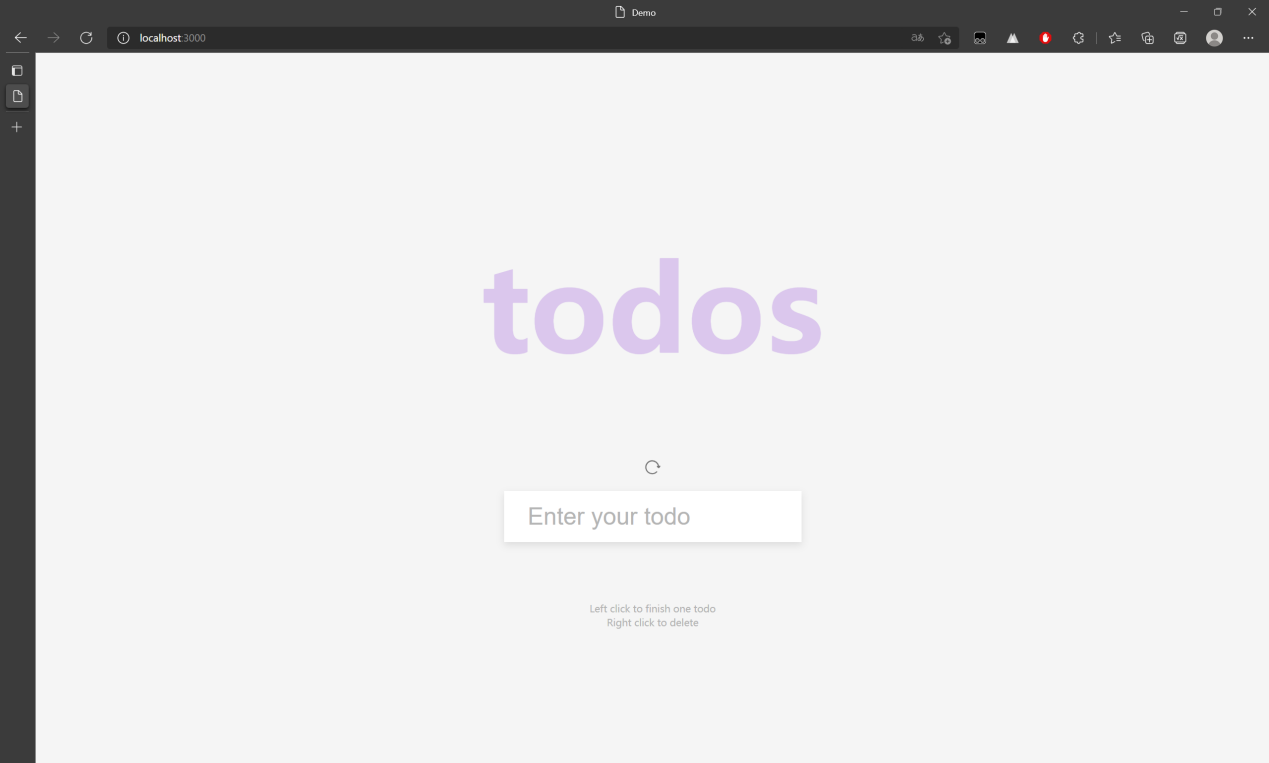


最终Jmeter主界面如下：



4.2对架构Demo系统进行性能测试（批量添加数据）

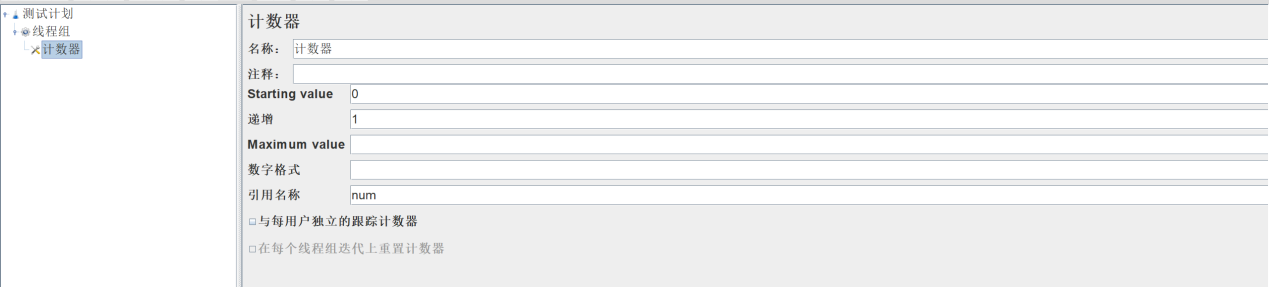
1.运行架构Demo系统，在浏览器中打开<http://localhost:3000/>查看系统是否正常启动



2.在测试计划中添加取样器中的”线程组”，设置线程数为1000，Ramp-Up时间为5秒



3.为了保证数据库主键的唯一性，并使生成的数据有规律，便于查看，在创建的线程组下添加配置元件中的”计数器”，初始值设置为0，递增为1，并设置引用名称为num，就可以在其他元件中通过${num}获取计数器的值

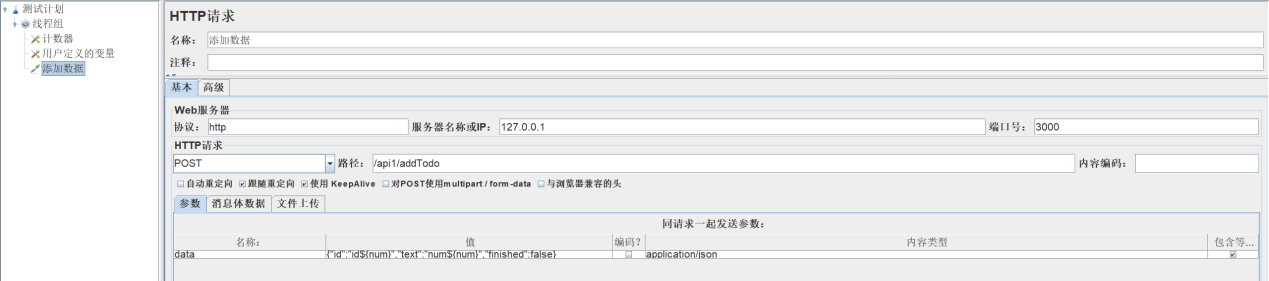


4.用户可以在线程组下添加配置元件中的”用户已定义的变量”，对需要测试的数据进行封装。添加一项，名称设置为id，值设置为test$num，此值可以作为添加数据的id值（本实验中未使用）

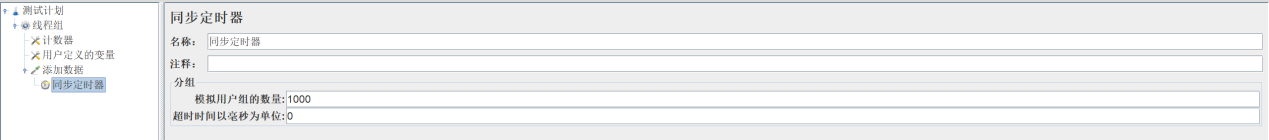
5.了解接口的相应参数，通过项目代码中的fe/my-app/src/App.jsx，查看到我们添加数据时使用post方法，调用了/${useApi}/addTodo接口，传递的数据为json格式，还传递了数据data，有三个属性分别为’id’,’text’,’finished’



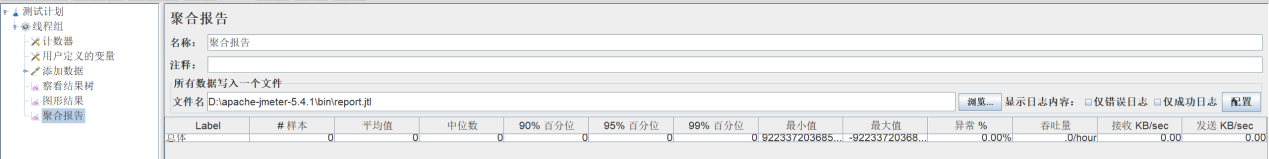
6.在线程组下添加取样器中的HTTP请求，根据上述信息填写web服务器、HTTP请求以及发送的参数名称和值，如果需要调用测试计划中定义的变量，需要以${名称}格式调用(如${num}，${id})。



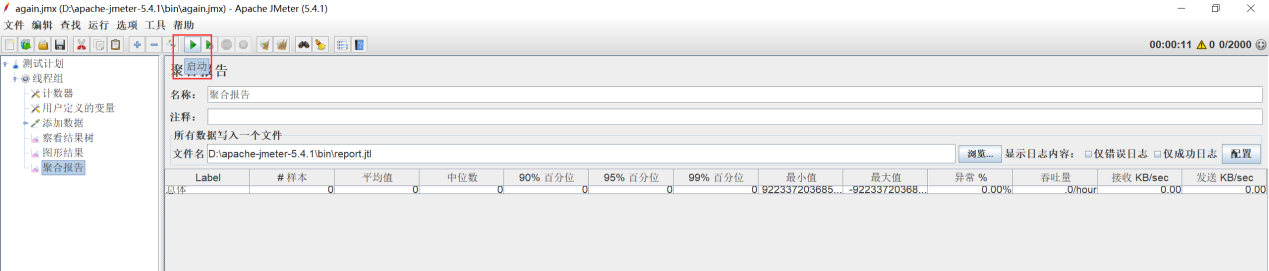
7.在HTTP请求下添加定时器中的”同步定时器（synchronized timer）”，设置模拟用户组的数量为1000，使1000个线程都生成后统一发送请求



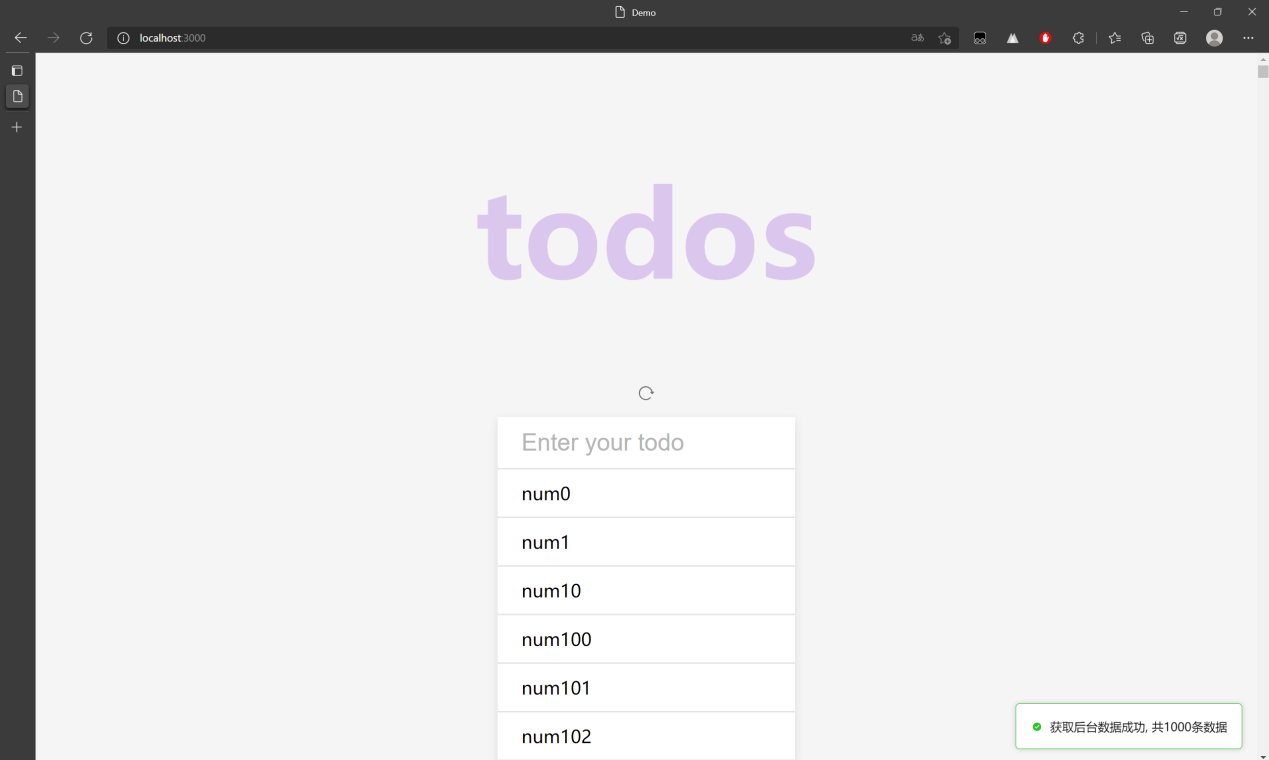
8.在线程组下分别添加监听器中的察看结果树、图形结果以及聚合报告，在聚合报告中填写一个文件名，使测试结果输出到文件中

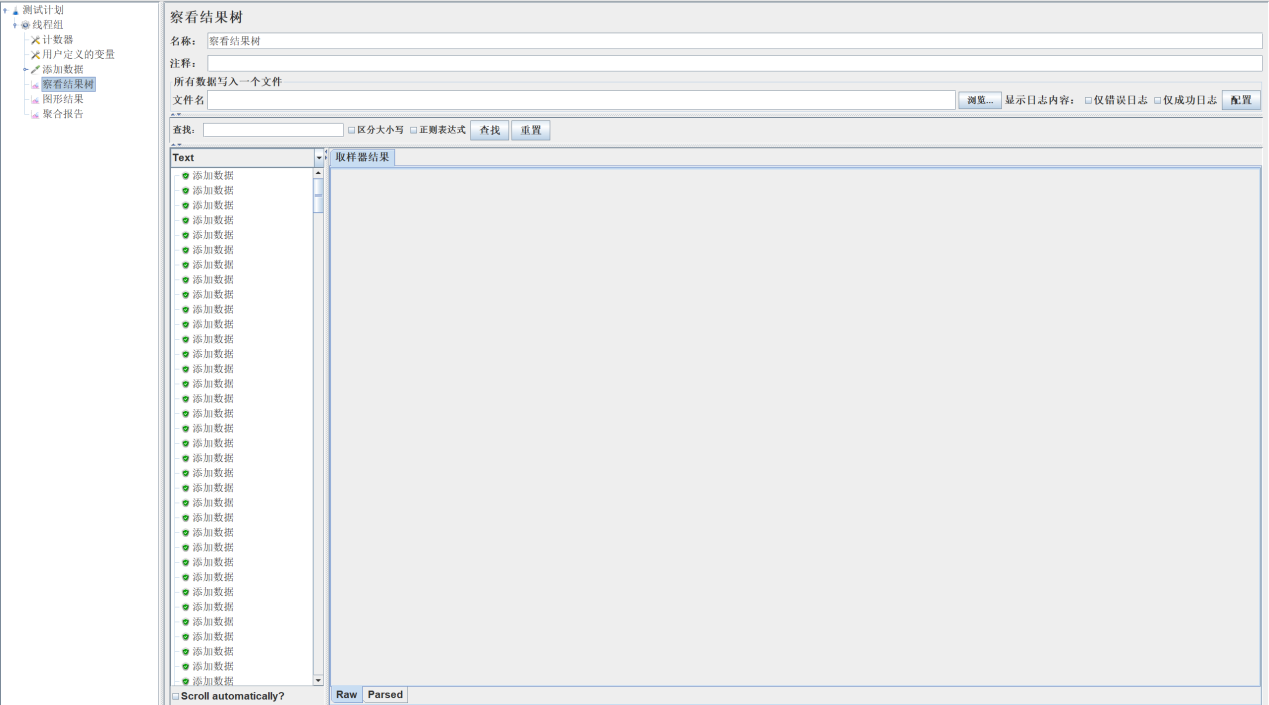


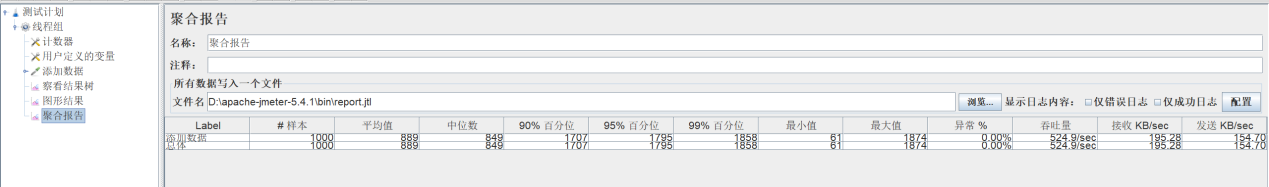
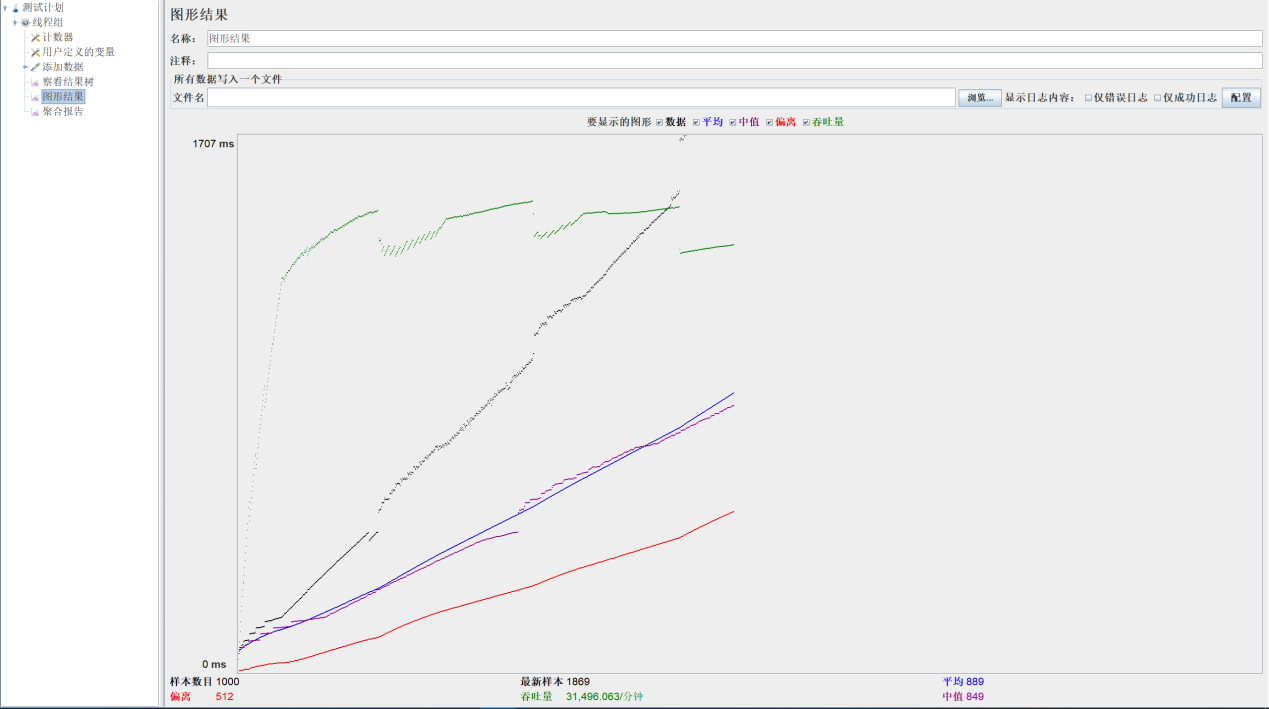
9.启动测试计划



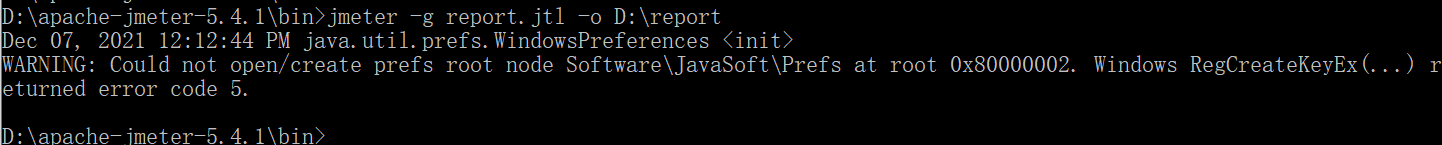
10.在网页、察看结果树、图形结果、聚合报告中查看测试结果



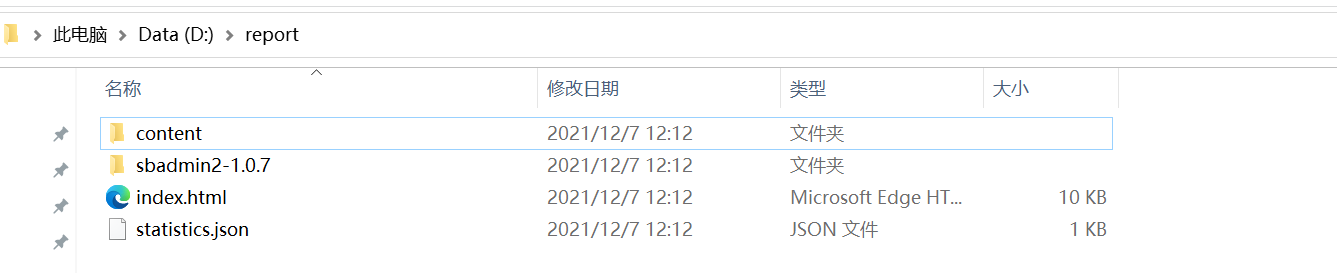


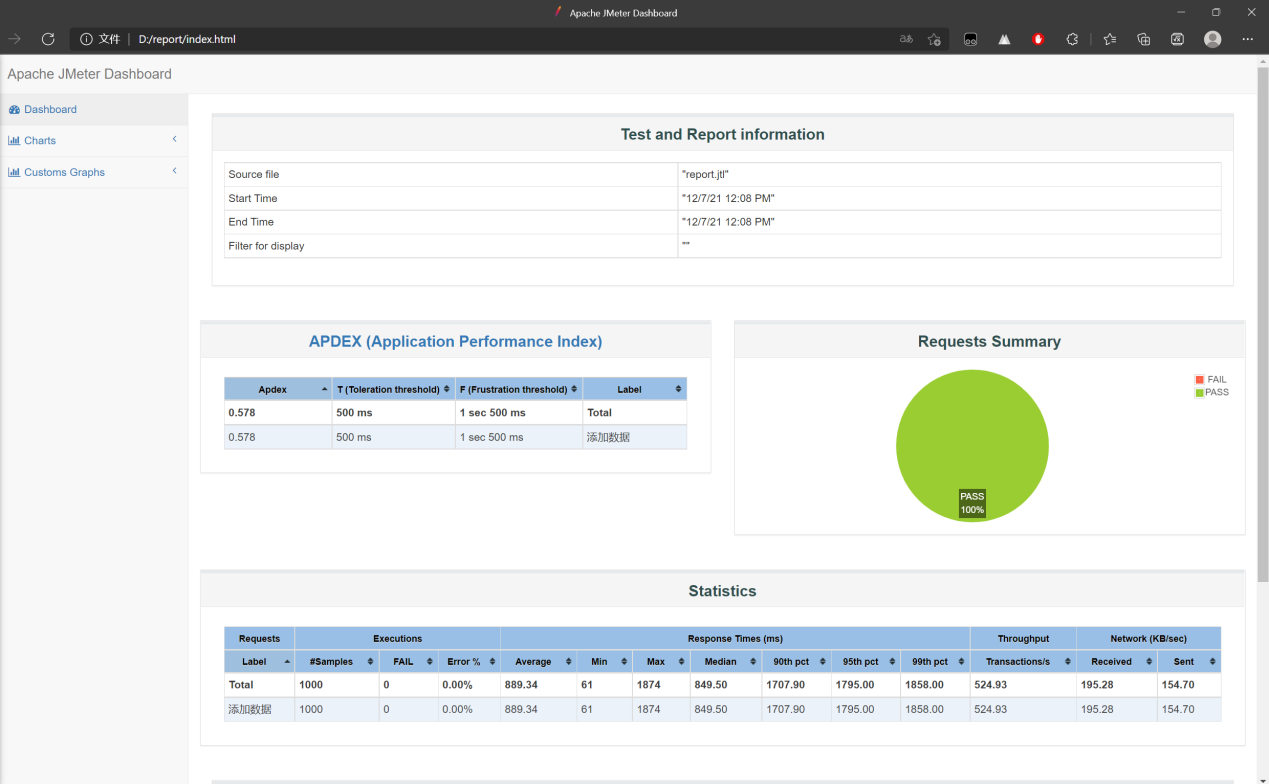


11.将生成的jtl测试结果文件转换成html，使用命令行前往jmeter安装路径下的bin文件夹中，执行jmeter -g 生成的jtl文件 -o 输出路径（输出路径为一个空的文件夹）



执行命令后，即可在输出路径看到结果，打开index.html即可看到全面的测试报告

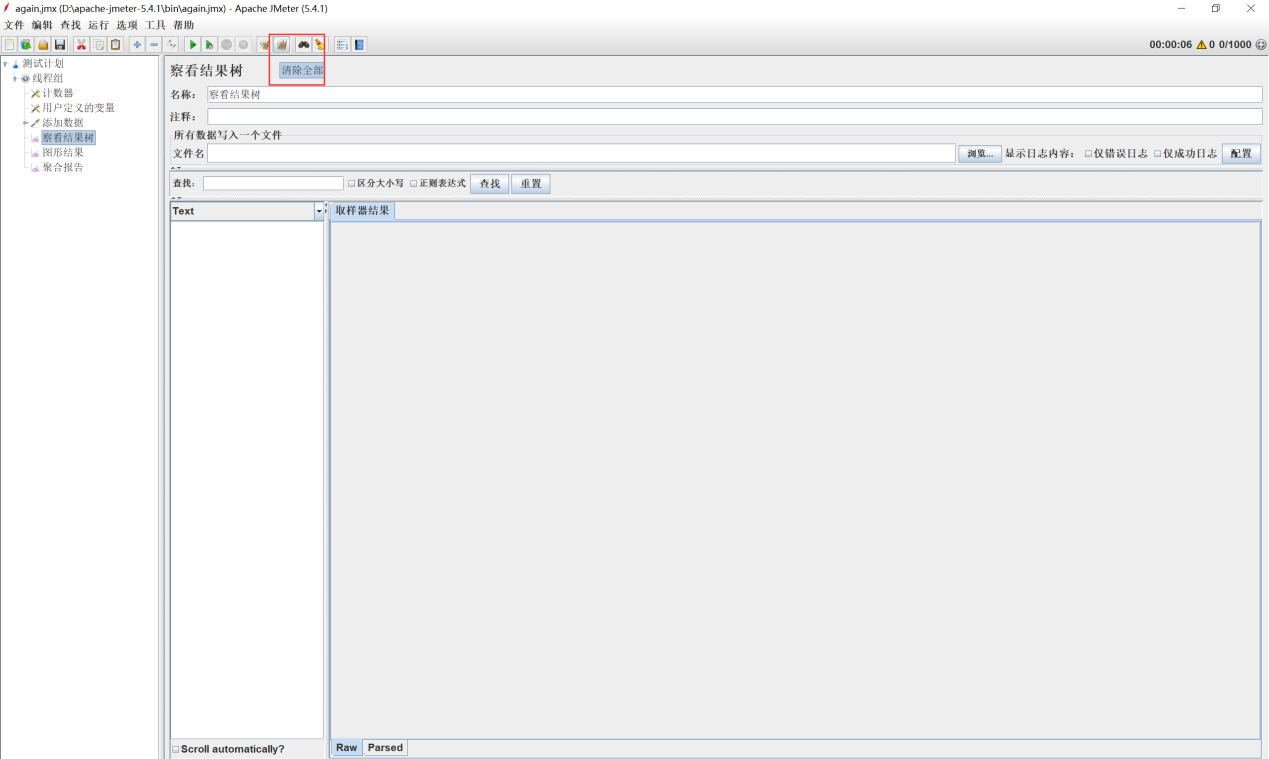




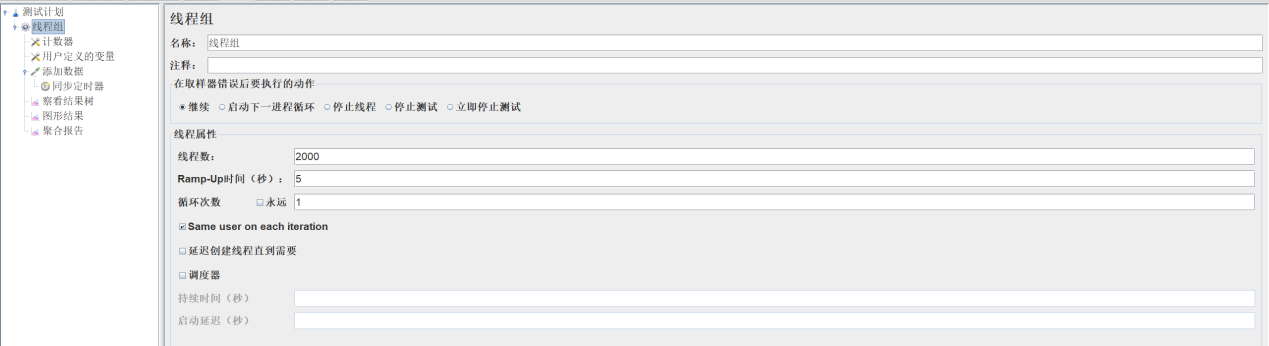
4.3对异常测试情况的判断

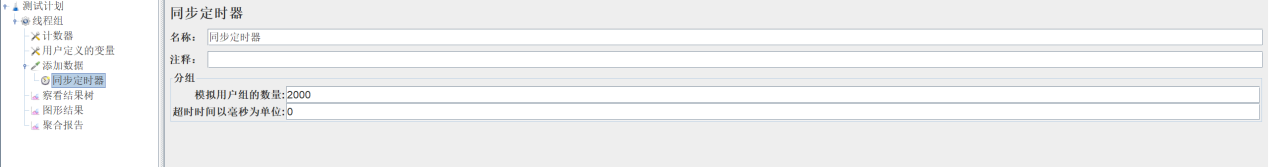
1.在数据库中对上一步骤生成的数据全部删除

2.对上一步骤的实验结果进行清除

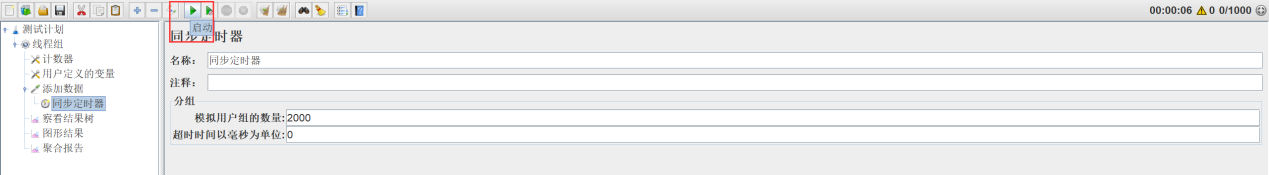


3.修改线程组的线程数为2000，同步定时器模拟用户组的数量为2000

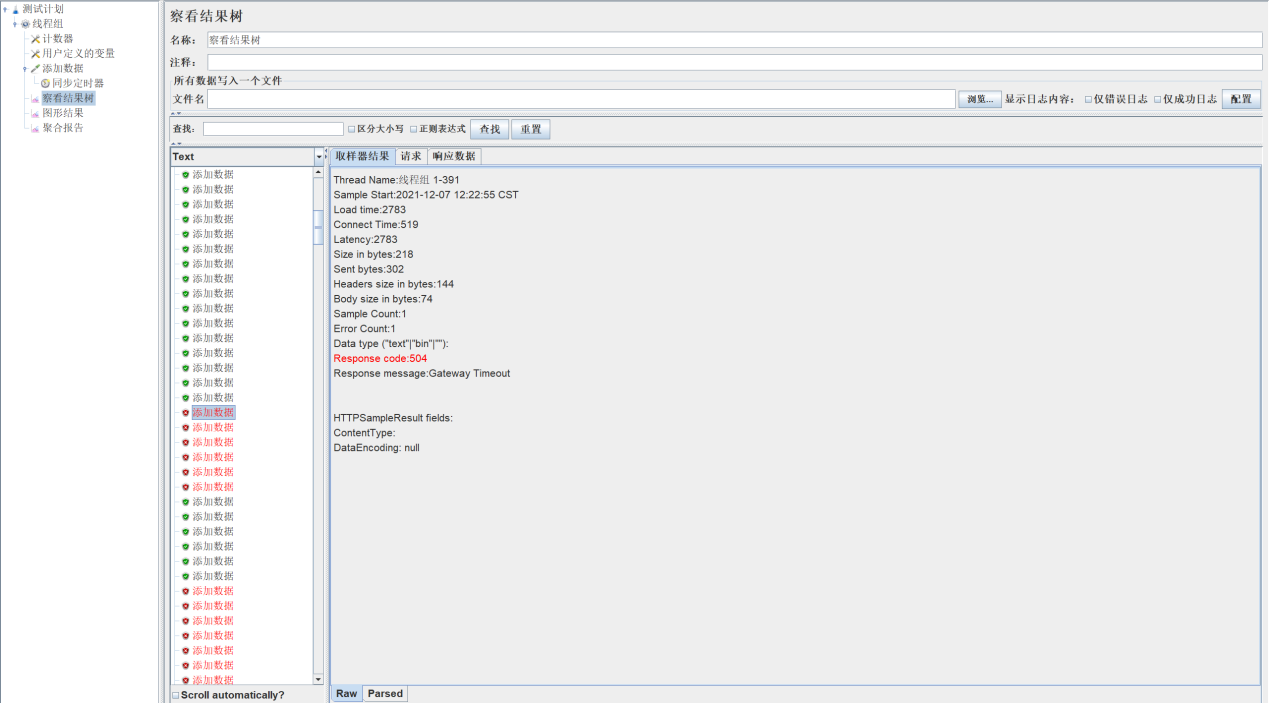




4.启动测试



5.查看察看结果树中，发现有失败请求，由于并发量太大，使得部分请求超时，直接返回504错误，可见并发请求数已经达到服务器的瓶颈。



**实验成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**