

華東理工大學

信息科学与工程学院

《软件工程》 实验报告九

系 别 计算机系

专 业 计算机科学与技术

年 级 2020 级

姓 名 刘子言

指导教师 阮 彤

2022-2023 学年 第 1 学期

实验九 性能测试

一、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。

二、实验环境

个人 PC 机，架构 Demo 系统，Jmeter5.5

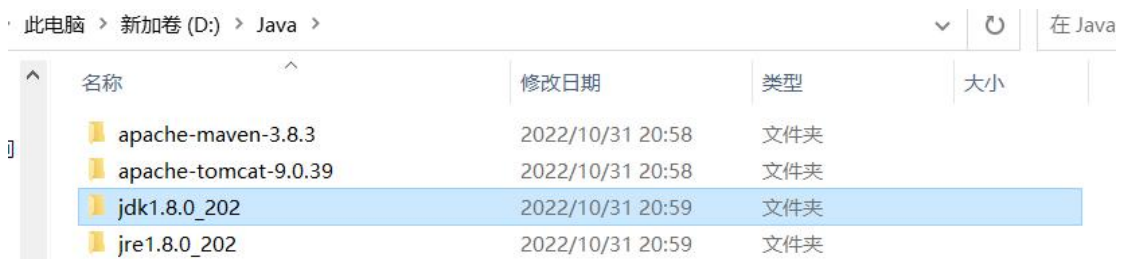
三、实验内容

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。



四、实验步骤

1、Jmeter 的安装和运行

(1) 安装 jdk (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：D:\JMeter\apache-jmeter-5.5



About

- Overview
- License

Download

- Download Releases
- Release Notes

Documentation

- Get Started
- User Manual
- Best Practices
- Component Reference
- Functions Reference
- Properties Reference
- Change History
- Javadocs
- JMeter Wiki
- FAQ (Wiki)



Tutorials

- Distributed Testing
- Recording Tests
- JUnit Sampler
- Access Log Sampler
- Extending JMeter

Community

- Issue Tracking
- Security
- Mailing Lists
- Source Repositories
- Building and Contributing
- Project info at Apache
- Contributors

Download Apache JMeter



We recommend you use a mirror to download our release builds, but you **must** [verify the integrity](#) of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from all the mirrors.

You are currently using <https://d1cdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backup mirrors](#) (at the end of the mirrors list) that should be available.

Other mirrors: [Change](#)

The **KEYS** link links to the code signing keys used to sign the product. The **PGP** link downloads the OpenPGP compatible signature from our main site. The **SHA-512** link downloads the sha512 checksum from the main site. Please [verify the integrity](#) of the downloaded file.

For more information concerning Apache JMeter, see the [Apache JMeter site](#).

[KEYS](#)

Apache JMeter 5.5 (Requires Java 8+)

Binaries

[apache-jmeter-5.5.lgz sha512 pgp](#)
[apache-jmeter-5.5.zip sha512 pgp](#)

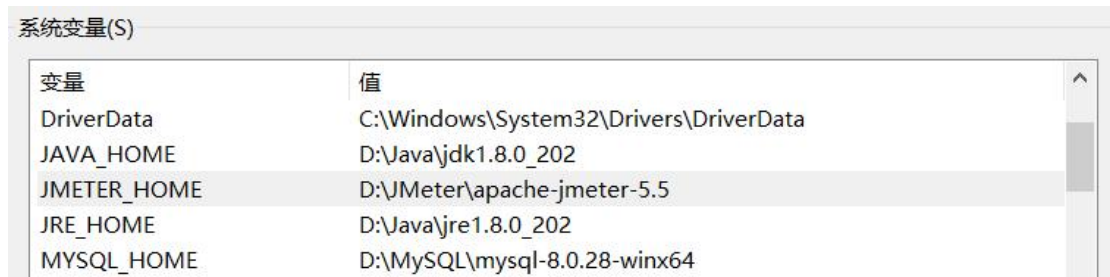
Source

[apache-jmeter-5.5_src.lgz sha512 pgp](#)
[apache-jmeter-5.5_src.zip sha512 pgp](#)

(3) 配置环境变量

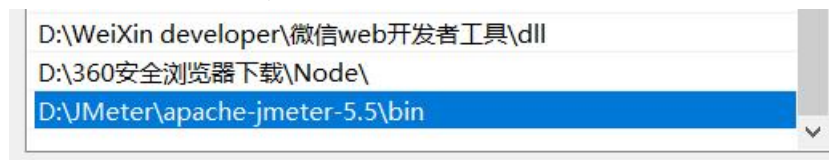
- 设置 jmeter 解压目录的 JMETER_HOME 环境变量

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



- 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

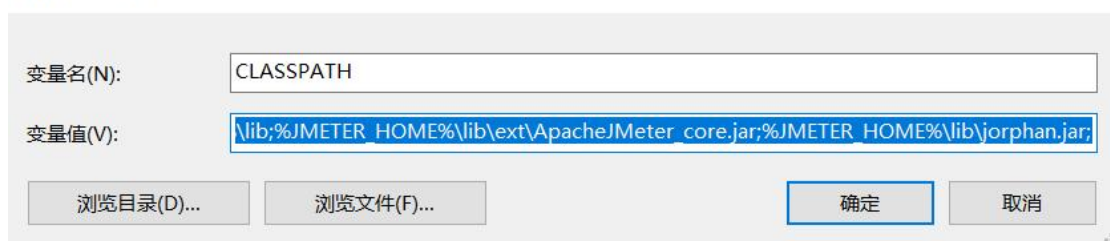


- 设置 jmeter 的 classpath 变量

系统属性中找到一个变量名为 CLASSPATH 的系统变量，新增两条：

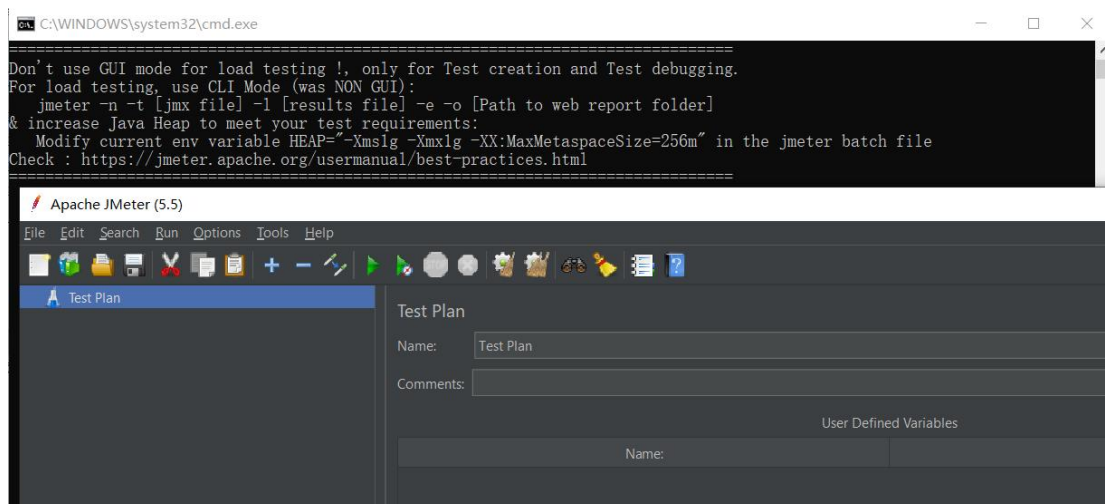
%JMETER_HOME%\lib\ext\ApacheJMeter_core.jar; %JMETER_HOME%\lib\jorphan.jar;

编辑系统变量



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



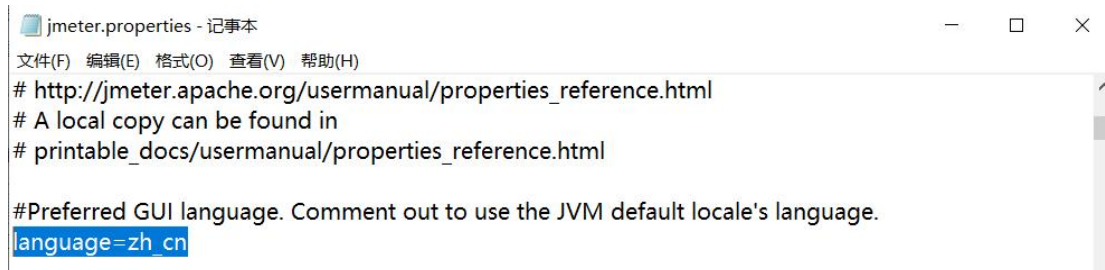


上图可见，jmeter 启动成功！

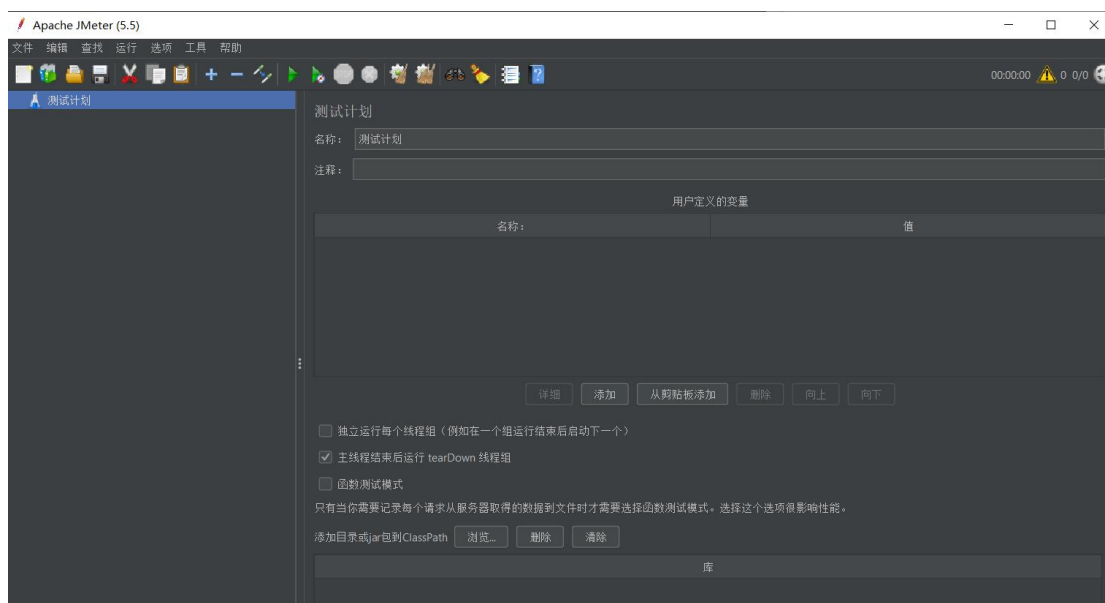
（5）将软件语言改为汉语

- 修改配置文件 jmeter.properties

通过记事本打开该文件，找到第 39 行修改以下字段：language = zh_cn



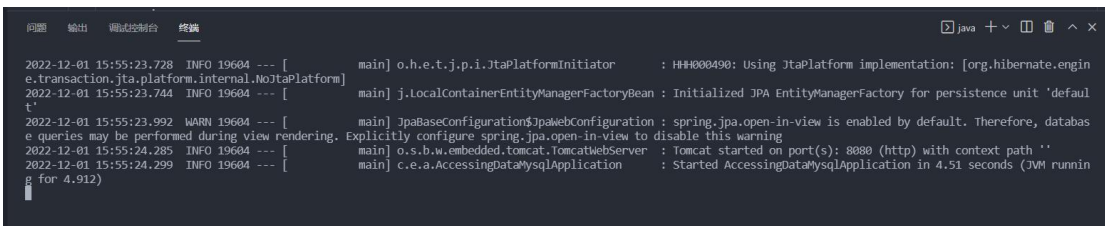
- 随后重新启动，最终 Jmeter 主界面如下：



2、对架构 restfulDemo 系统进行性能测试（批量添加数据）

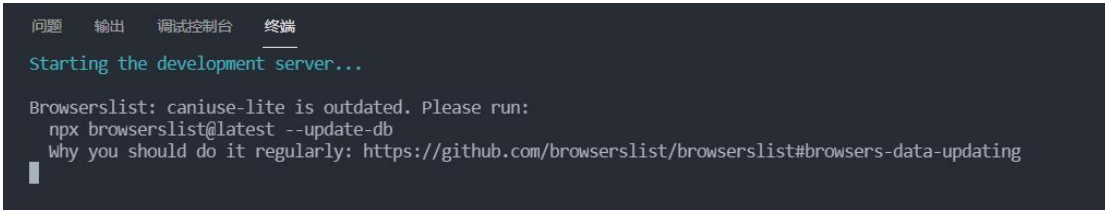
（1）运行架构 restfulDemo 系统

- 启动后端，在...\restfulDemo\be-spring 目录下执行 mvn spring-boot:run 命令，启动成功：



```
2022-12-01 15:55:23.728 INFO 19604 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-12-01 15:55:23.744 INFO 19604 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-12-01 15:55:23.992 WARN 19604 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2022-12-01 15:55:24.285 INFO 19604 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-12-01 15:55:24.299 INFO 19604 --- [main] c.e.a.AccessingDataMysqlApplication : Started AccessingDataMysqlApplication in 4.51 seconds (JVM running for 4.912)
```

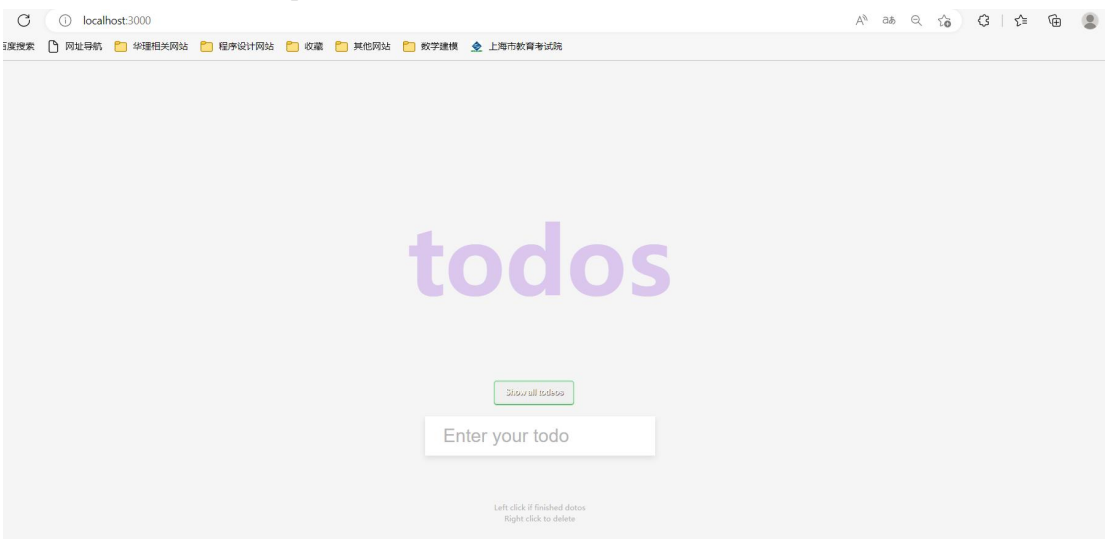
- 启动前端，在...\restfulDemo\fe\my-app 目录下执行 npm start



```
Starting the development server...

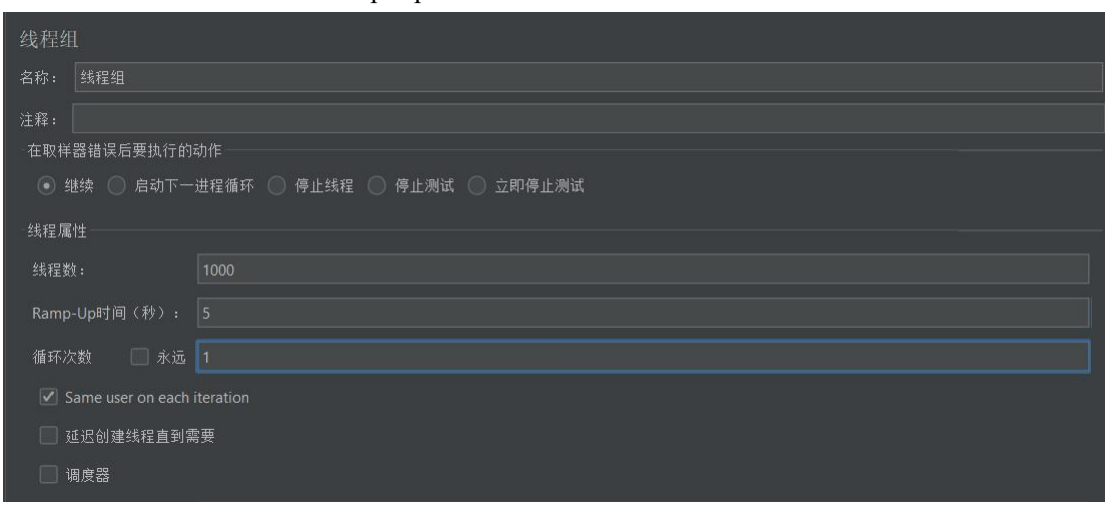
Browserslist: caniuse-lite is outdated. Please run:
  npx browserslist@latest --update-db
  why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
```

在浏览器中打开 http://localhost:3000/，确认系统已经正常启动：



（2）在测试计划中添加取样器中的“线程组”

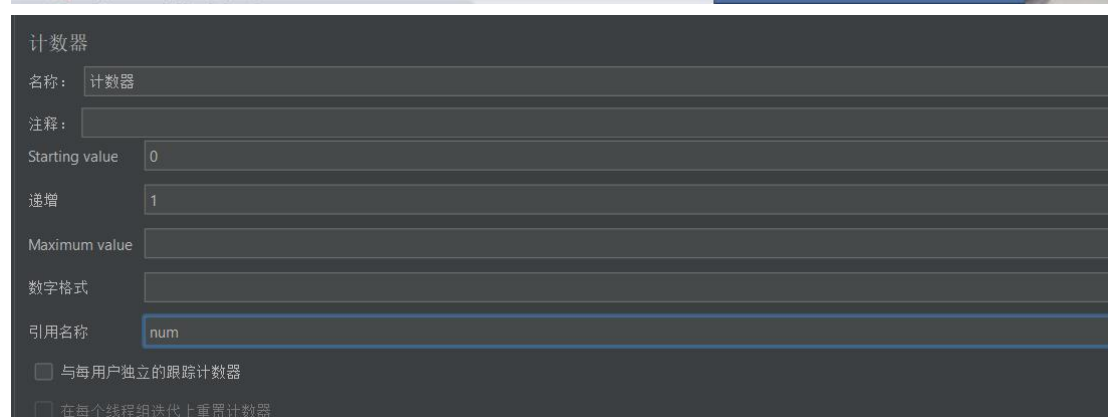
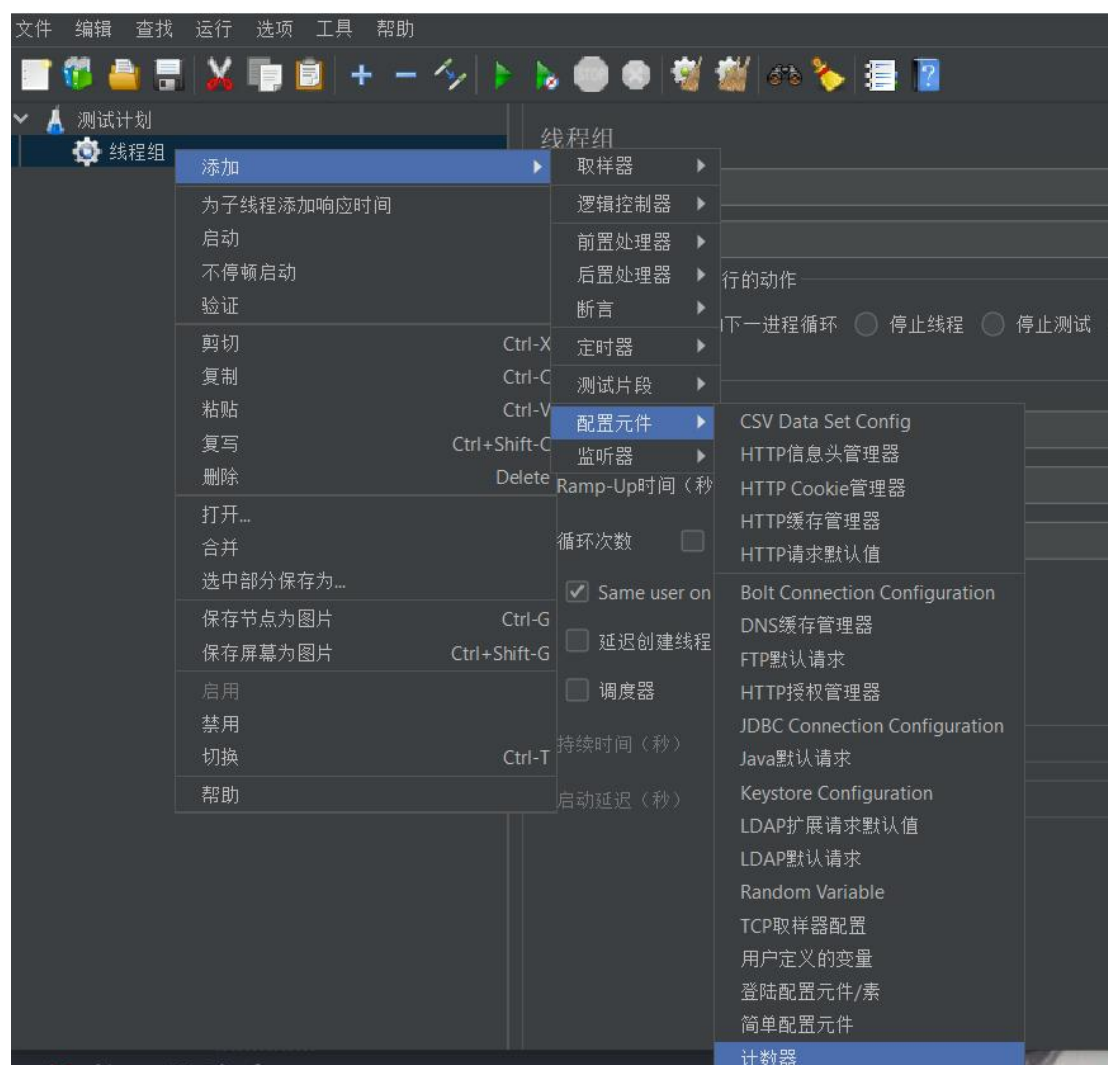
设置线程数为 1000，Ramp-Up 时间为 5s：



The screenshot shows the 'Thread Group' configuration dialog in JMeter. The 'Name' field is set to '线程组'. The 'Notes' field is empty. Under 'Action to perform when sampler errors occur', the 'Continue' radio button is selected. In the 'Thread Properties' section, the 'Thread count' is set to '1000', the 'Ramp-Up time (s)' is set to '5', and the 'Loop count' is set to '1'. The 'Same user on each iteration' checkbox is checked. The 'Delay thread creation until needed' and 'Scheduler' checkboxes are unchecked.

（3）添加配置元件“计数器”

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）添加配置元件“用户已定义的变量”

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（但在

本实验中没有用到)

用户定义的变量

名称：

用户定义的变量

注释：

用户定义的变量

名称：	值	描述
id	test\$num	添加数据的id值

(5) 了解接口的相应参

通过项目代码中的 fe/my-app/src/App.jsx，可以看到：

添加数据时使用的是 post 方法，调用了/\${useApi}/addTodo 接口；传递了数据 data，数据格式为 json，它有三个属性，分别为 'id'，'text'，'finished'。

```
46 // check in frontend, input can not be empty 检查数据是否合法（且不能为空）
47 if (inputVal) {
48   const todoObj = {
49     'id': generateId().toString(),
50     'text': inputVal.toString(),
51     'finished': false
52   }
53   // post to backend 将数据送到后端
54   axios.post(`${useApi}/addTodo`, {
55     headers: {'Content-Type': 'application/json'},
56     data: JSON.stringify(todoObj)
57   }).then((res) => {
```

(6) 添加取样器中的 HTTP 请求

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，以\${名称}格式调用（例如\${num}，\${id}）

HTTP请求

名称：

HTTP请求

注释：

基本 高级

Web服务器

协议：

http

 服务器名称或IP：

127.0.0.1

 端口号：

3000

HTTP请求

POST 路径：

/api2/addTodo

 内容编码：

☐ 自动重定向 ☒ 跟随重定向 ☒ 使用 KeepAlive ☐ 对POST使用multipart / form-data ☐ 与浏览器兼容的头

参数 消息体数据 文件上传

同请求一起发送参数：

名称：	值	编码？	内容类型	包含等于？
data	{"id":"id\${num}","text":"num\${num}","finish...	<input type="checkbox"/>	application/json	<input checked="" type="checkbox"/>

HTTP请求

名称: HTTP请求

注释:

基本 高级

Web服务器

协议: http 服务器名称或IP: 127.0.0.1 端口号: 3000

HTTP请求

POST 路径: /api2/addTodo 内容编码:

☐ 自动重定向 ☒ 跟随重定向 ☒ 使用 KeepAlive ☐ 对POST使用multipart / form-data ☐ 与浏览器兼容的头

参数 消息体数据 文件上传

同请求一起发送参数:

名称:	值	编码?	内容类型	包含等于?
data	{\ "id":\ "id\${num}\",\ "text":\ "num\${num}\",\ ...	<input type="checkbox"/>	application/json	<input checked="" type="checkbox"/>

HTTP请求

名称: HTTP请求

注释:

基本 高级

Web服务器

协议: http 服务器名称或IP: 127.0.0.1 端口号: 3000

HTTP请求

POST 路径: /api2/addTodo 内容编码:

☐ 自动重定向 ☒ 跟随重定向 ☒ 使用 KeepAlive ☐ 对POST使用multipart / form-data ☐ 与浏览器兼容的头

参数 消息体数据 文件上传

```
1 {"headers":{"Content-Type":"application/json"},"data":
2 [{"id":"id${num}","text":"num${num}","finished":false}]0
```

（7）添加定时器中的“同步定时器”

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 200：

同步定时器

名称: 同步定时器

注释:

分组

模拟用户组的数量: 200

超时时间以毫秒为单位: 0

（8）添加监听器中的察看结果树、图形结果以及聚合报告

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

聚合报告

名称: 聚合报告

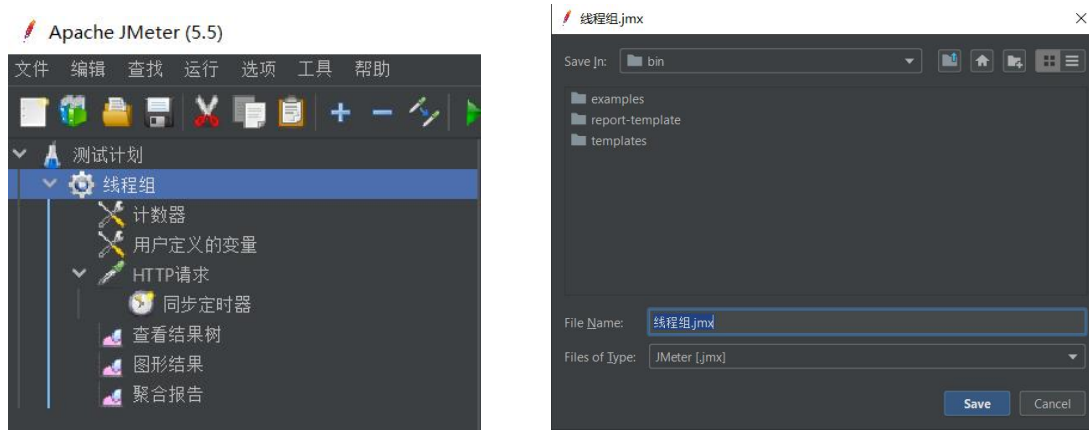
注释:

所有数据写入一个文件

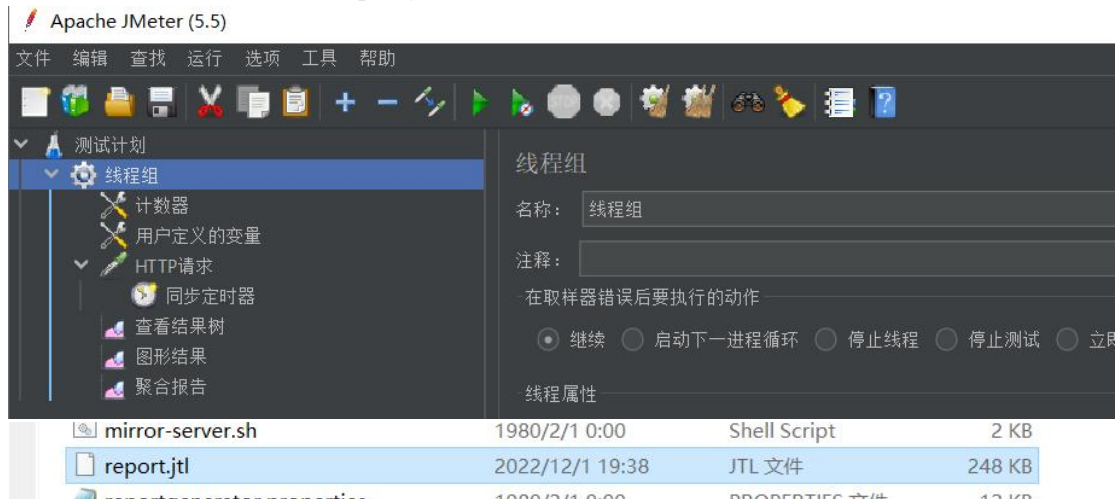
文件名: D:\JMeter\apache-jmeter-5.5\bin\report.jtl 浏览... 显示日志内容: ☐ 仅错误日志 ☐ 仅成功日志 配置

Label	# 样本	平均值	中位数	90% 百分...	95% 百分...	99% 百分...	最小值	最大值	异常 %	吞吐量	接收 KB/s...	发送 KB/s...
总体	0	0	0	0	0	0	92233720...	-9223372...	0.00%	.0/hour	0.00	0.00

到目前为止，测试计划中线程组目录如下：

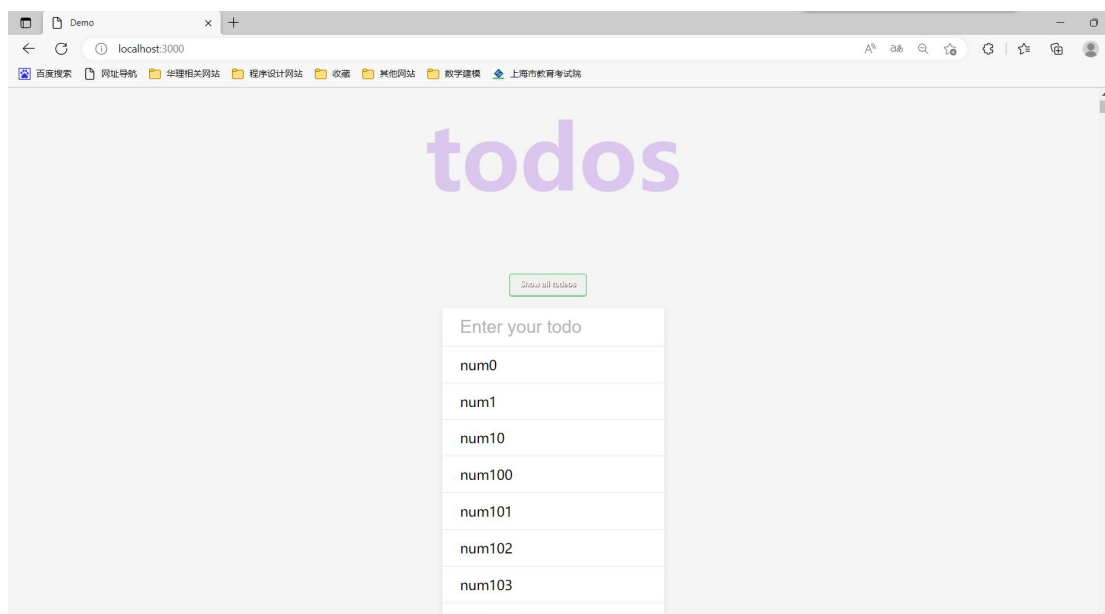


(9) 启动测试计划，生成 report.jtl 文件



(10) 在网页、查看结果树、图形结果、聚合报告中查看测试结果

• 网页获取后台数据的结果以及数据库中数据的结果



- 聚合报告中的测试结果

聚合报告

名称: 聚合报告

注释:

所有数据写入一个文件

文件名: D:\JMeter\apache-jmeter-5.5\bin\report.jtl 显示日志内容: ☐ 仅错误日志 ☐ 仅成功日志

Label	# 样本	平均值	中位数	90% 百分...	95% 百分...	99% 百分...	最小值	最大值	异常 %	吞吐量	接收 KB/s...	发送 KB/s...
HTTP请求	1000	415	414	662	703	740	95	748	0.00%	224.3/sec	41.84	63.92
总体	1000	415	414	662	703	740	95	748	0.00%	224.3/sec	41.84	63.92

(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, 执行以下命令
jmeter -g report.jtl -o D:\JMeter\report

```
管理员: C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.2251]
(c) Microsoft Corporation。保留所有权利。

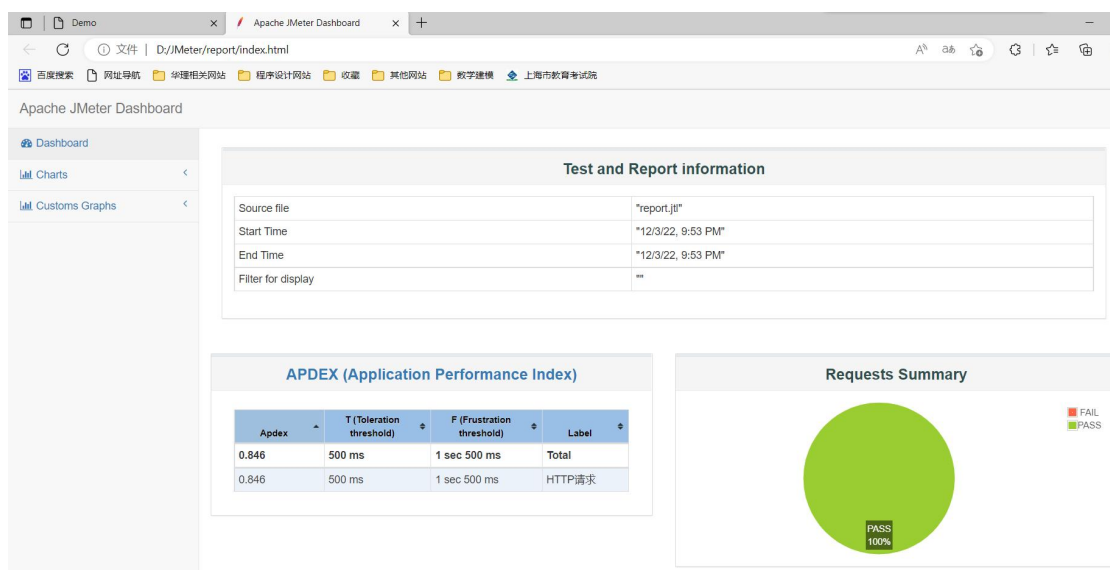
D:\JMeter\apache-jmeter-5.5\bin>jmeter -g report.jtl -o D:\JMeter\report
```

- 执行命令后, 在 report 文件夹中看到结果:

此电脑 > 新加卷 (D:) > JMeter > report

名称	修改日期	类型	大小
content	2022/12/3 22:07	文件夹	
sbadmin2-1.0.7	2022/12/3 22:07	文件夹	
index.html	2022/12/3 22:07	Microsoft Edge ...	10 KB
statistics.json	2022/12/3 22:07	JSON 源文件	1 KB

- 打开 index.html 看到全面的测试报告:



Statistics													
Requests		Executions			Response Times (ms)							Throughput	Network (KB/sec)
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	1000	0	0.00%	415.28	95	748	414.00	663.80	703.00	741.98	224.32	41.84	63.92
HTTP请求	1000	0	0.00%	415.28	95	748	414.00	663.80	703.00	741.98	224.32	41.84	63.92

Errors			
Type of error	Number of errors	% in errors	% in all samples

Top 5 Errors by sampler													
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error
Total	1000	0											

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数量为 2000

线程组

名称：

线程组

注释：

在取样器错误后要执行的动作

继续

启动下一进程循环

停止线程

停止测试

立即停止测试

线程属性

线程数：

2000

Ramp-Up时间（秒）：

5

循环次数

永远

1

Same user on each iteration

延迟创建线程直到需要

调度器

同步定时器

名称：

同步定时器

注释：

分组

模拟用户组数量：

2000

超时时间以毫秒为单位：

0

- (4) 启动测试，聚合报告结果如下，异常率为 86.45%：

聚合报告

名称：

聚合报告

注释：

所有数据写入一个文件

文件名：

D:\JMeter\apache-jmeter-5.5\bin\report.jtl

浏览...

显示日志内容：

仅错误日志

仅成功日志

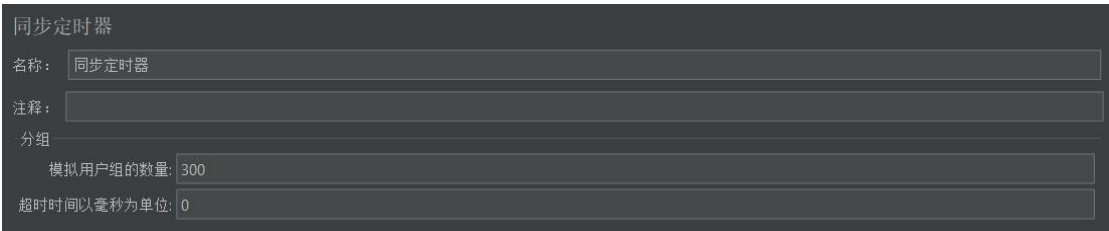
配置

Label	# 样本	平均值	中位数	90% 百分...	95% 百分...	99% 百分...	最小值	最大值	异常 %	吞吐量	接收 KB/s...	发送 KB/s...
HTTP请求	2000	105	1	609	868	987	0	1000	86.45%	1949.3/sec	4318.18	75.11
总体	2000	105	1	609	868	987	0	1000	86.45%	1949.3/sec	4318.18	75.11

(5) 查看察看结果树中，发现有失败请求，报错 504，可能是由于并发量太大，使得部分请求超时，可见并发请求数已经达到服务器的瓶颈。

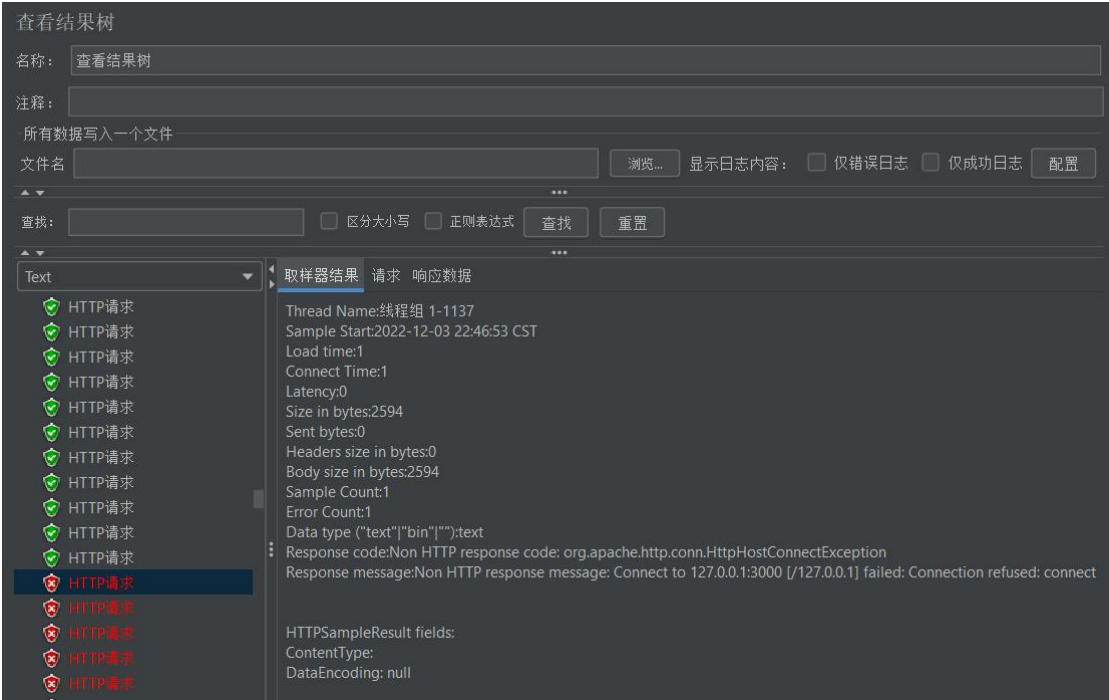


(6) 再次清除之前的测试结果数据，修改线程组的线程数为 1200，同步定时器模拟用户组的数量为 300，再次测试。



由测试结果可知异常率 19.75%，此时不再是 504 请求超时的报错，而是有部分数据异常 http 请求无响应，服务器返回的请求消息是拒绝连接：

Label	# 样本	平均值	中位数	90% 百分...	95% 百分...	99% 百分...	最小值	最大值	异常 %	吞吐量	接收 KB/s...	发送 KB/s...
HTTP请求	1200	373	390	707	741	770	0	781	19.75%	287.7/sec	187.00	65.85
总体	1200	373	390	707	741	770	0	781	19.75%	287.7/sec	187.00	65.85



五、实验心得

通过本次的实验，我了解了 Jmeter 性能测试工具的基本用途以及简单的操作，掌握了 Jmeter 性能测试工具的基本测试过程，并且能够使用 Jmeter 对项目进行负载测试。

在实验中，我在 restfulDemo 架构系统上进行了测试脚本的编写以及性能测试，对测试结果进行了分析，并尝试通过修改 Jmeter 线程组中的部分参数（线程数、模拟用户组的数量等）来加深对异常测试情况的了解、判断，发现其原因。

此次实验我的收获颇丰，同时这也为我在之后的项目推进及性能测试方面打下了实践基础。

实验成绩: _____ 教师签名: _____