一、实验目的

1. 验证所学理论，巩固所学知识并加深理解；
2. 培养学生设计负载阶梯测试分析的能力；
3. 熟悉 JMeter 中添加和管理插件的基本操作；
4. 熟悉 JMeter 中设置负载阶梯性能测试的过程和操作.

二、实验项目内容

(一) 以 www.baidu.com 为例，建立对百度首页的访问。

要求：从该场景开始，加入 500 用户同时访问该场景，同时测试 20 分钟（20 分钟为压力测试时间）。然后分别绘制相应的步进曲线图和聚合报告。

1. 开始时模拟 50 用户访问，然后每 10 秒增加 25 用户，直至达到 500 用户；

2. 峰值等待 2 分钟，然后每 10 秒减少 50 用户，直至回到 50 用户并发；

3. 提交相关设置截图；

4. 为该场景增加“结果树”报告和“聚合报告”，提交相关截图；

5. 运行完成后，截图聚合报告的图形化内容（含平均响应时间、95%、99%和最大响应时间），并以文字方式提交“平均响应时间”和“最大响应时间”。

(二) 自定义一个动态网站，可以只含首页，然后用步进方式进行测试并见监看服务器性能。

1. 该网站使用开发语言不限。

(1) 有简单的显示界面；

(2) 允许用 Get 方式提交参数 Name，值为 1 到 100000 之间的随机数；

(3) 服务器将收到 Name 写入到 Session 中。

2. 从该场景开始，加入 500 用户同时访问该场景，同时测试 20 分钟（20 分钟为压力测试时间）。然后分别绘制相应的步进曲线图和聚合报告。

(1) 开始时模拟 50 用户访问，然后每 5 秒增加 20 用户，直至达到 500 用户；

(2) 峰值等待 20 分钟，然后每 5 秒减少 50 用户，直至回到 50 用户并发；

(3) 提交相关设置截图；

(4) 为该场景增加“结果树”报告和“聚合报告”，提交相关截图

(5) 运行完成后，截图聚合报告的图形化内容（含平均响应时间、95%、99%和最大响应时间），并以文字方式提交“平均响应时间”和“最大响应时间”；

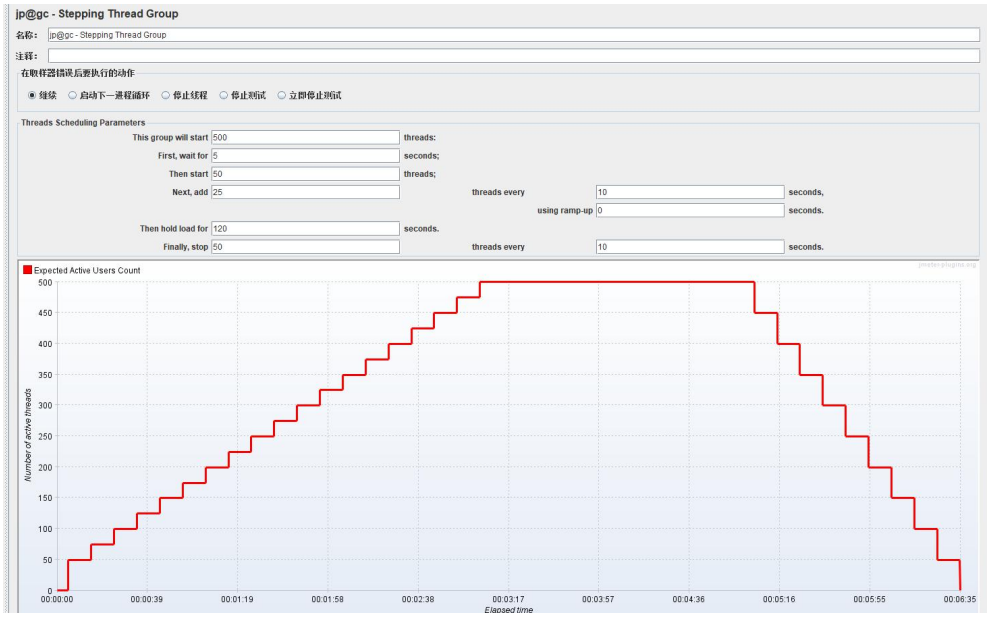
(6) 监看服务器性能，提交该服务器在 200 用户以下时的性能截图和 500 用户时的性能截图，以及减少到 50 用户时的性能截图。性能截图包括：CPU、内存、网络。

三、实验过程或算法（源程序）

1. 以 www.baidu.com 为例，建立对百度首页的访问。

1.1 配置阶梯负载测试参数

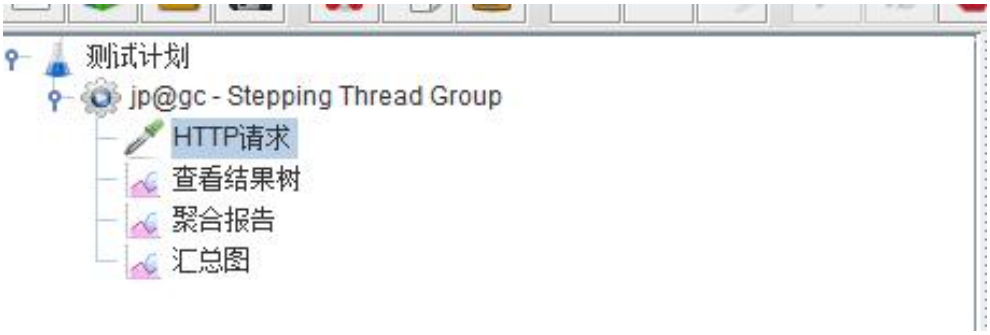
加入 500 线程模拟 500 用户同时访问该场景。等待 5s 期间没有用户访问，刚开始时模拟 50 用户访问。然后每 10s 增加 25 用户，峰值等待 2 分钟(120s)，然后每 10s 减少 50 用户，具体配置如下。



1.2 配置 HTTP 请求

配置 https 协议，服务器名称或 IP 为 www.baidu.com

1.3 增加结果树、聚合报告和汇总图



2. 自定义一个动态网站，可以只含首页，然后用步进方式进行测试并见监看服务器性能。

2.1 自定义服务器网站

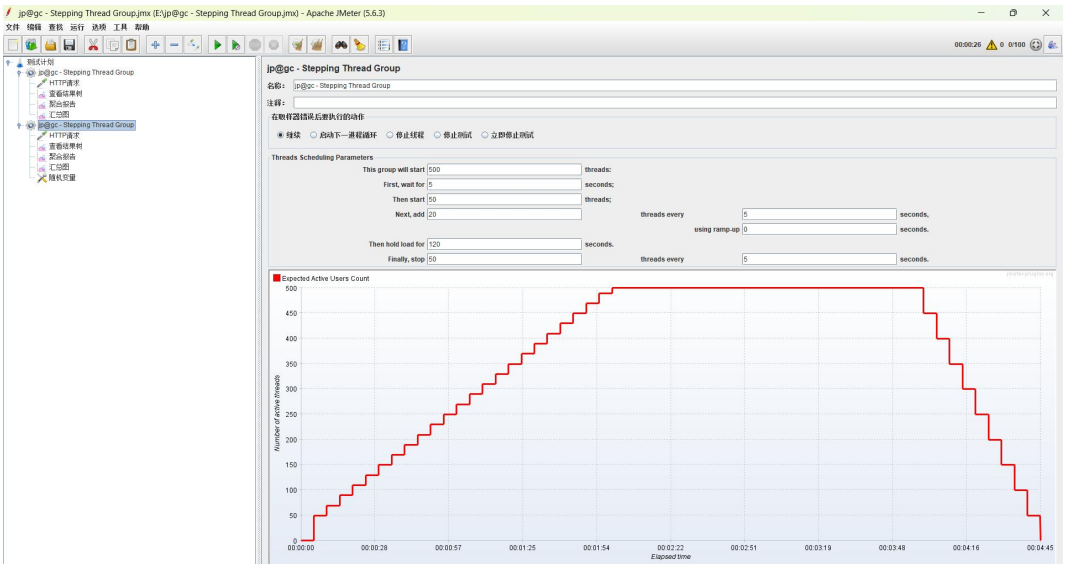
以 Python Flask 框架为基础，构造 127.0.0.1:7700/hello?Name=x 网页请求，其中 x 为用户传入的参数。

```python
from flask import Flask, request, session

app = Flask(__name__)

@app.route('/hello', methods=['GET'])
def hello():
    # 获取请求参数Name
    name = request.args.get('Name')
    # 将Name存入Session
    session['Name'] = name
    # 构造返回的字符串
    message = "hello world " + name if name else "hello world"
    # 返回响应
    return message

if __name__ == '__main__':
    app.run(debug=True, port=7700)
```

在浏览器中访问该 URL 效果如下：

2.2 配置阶梯负载测试参数

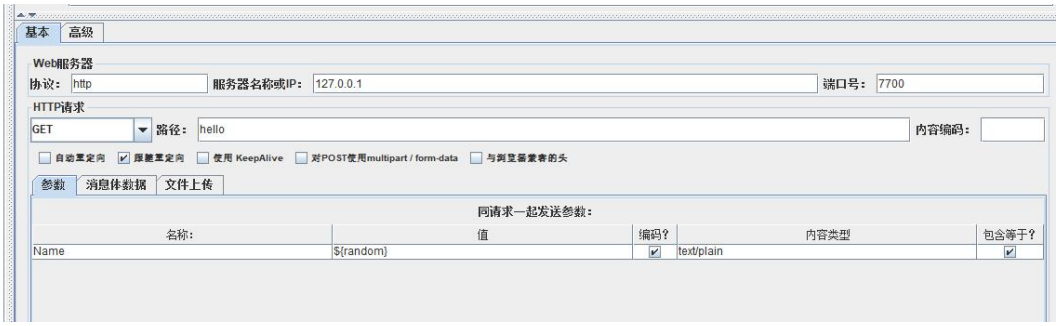
加入 500 线程模拟 500 用户同时访问该场景。等待 5s 期间没有用户访问，刚开始时模拟 50 用户访问。然后每 5s 增加 20 用户，峰值等待 2 分钟(120s)，然后每 5s 减少 50 用户，具体配置如下。



2.3 配置 HTTP 请求参数

设置 Name 随机变量，随机值在 1-100000 之间波动

使用 Name 作为传参，服务器为本地本地环回地址 127.0.0.1，端口号为 7700.



2.4 配置系统参数

在实际运行中，当用户数增加到 300 以上时，会产生较多的“Non HTTP response message: Address already in use: connect”错误导致异常率较高，经查阅资料[1]可知，需对注册表进行如下修改以降低异常率：

[1] 原文链接：https://blog.csdn.net/weixin_46504244/article/details/118072319

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters → 新建 DWORD 值：

name：MaxUserPort， value：65534（十进制）

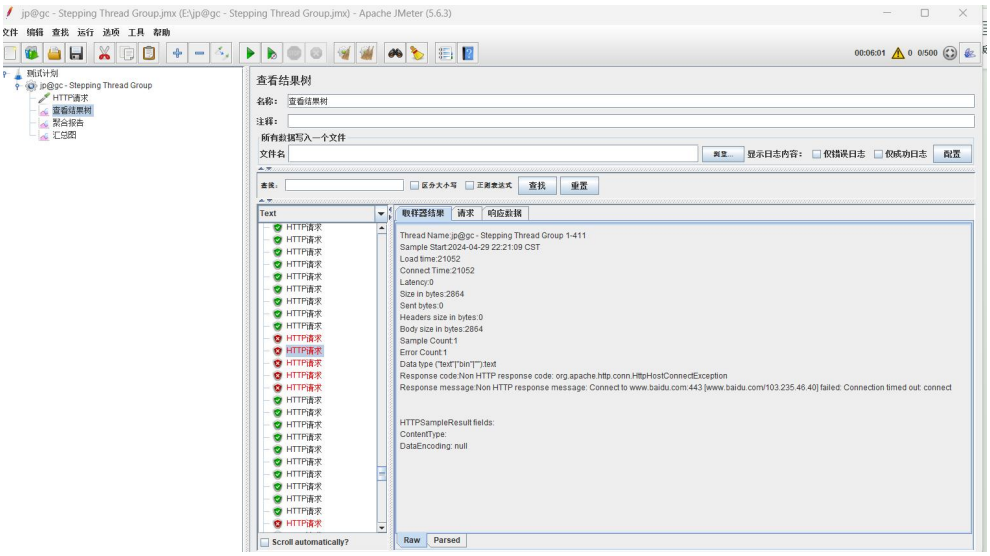
name：TcpTimedWaitDel，value：30（十进制）

四、实验结果及分析和（或）源程序调试过程

1. 以 www.baidu.com 为例，建立对百度首页的访问

1.1 查看结果树

可以看到部分网页请求失败，发生了 443 Time out 的错误，可能是网络原因或者代理造成的，但大部分请求是成功的。



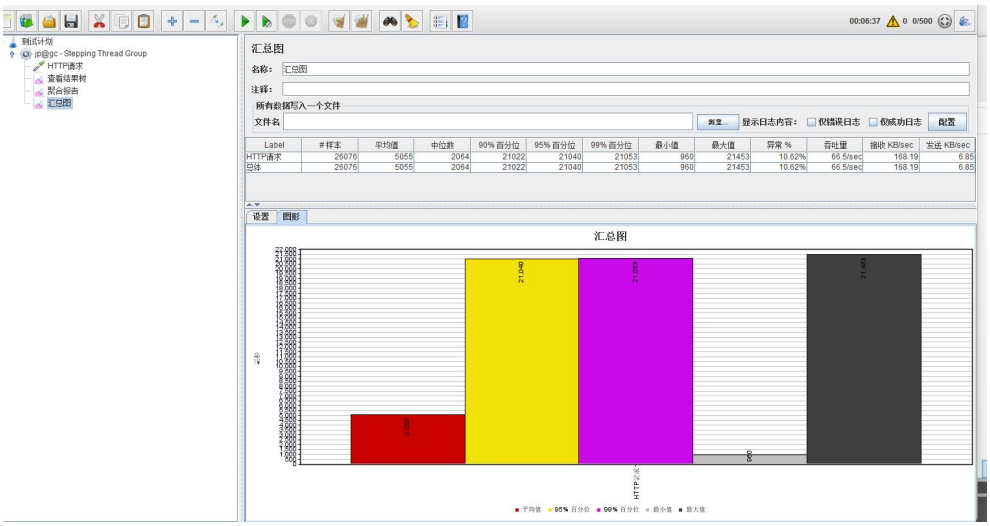
1.2 聚合报告和汇总图

实验发现 ram-up 为不同值时会得到不同的实验结果。其中 ram-up 时间越大，在一定程度上可以降低请求异常率。

- **当 ram-up = 0 时，异常率为 10.62%.**

平均响应时间：5055

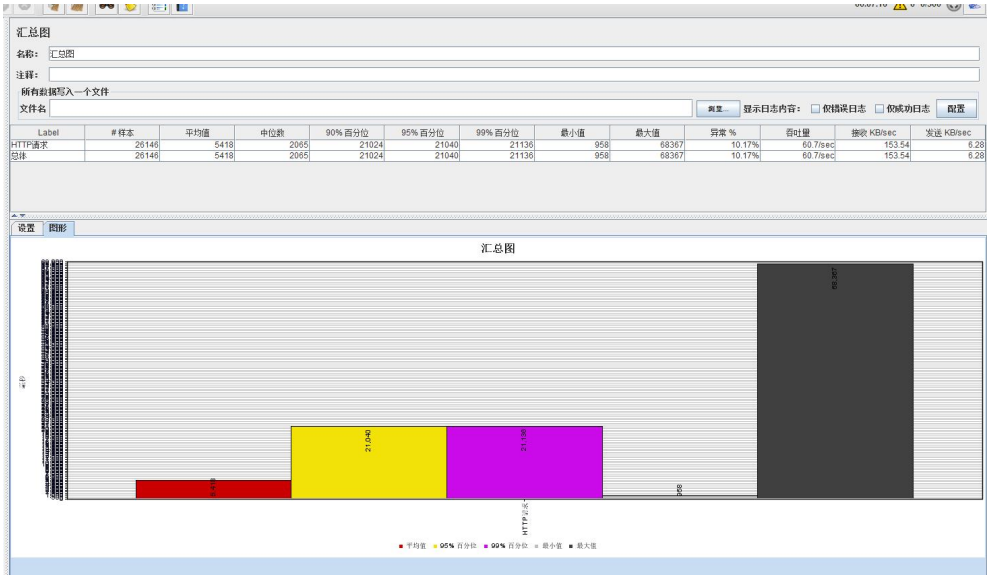
最大响应时间：21453.



- 当 **ram-up = 2 时，异常率为 10.17%.**

  平均响应时间：5418

  最大响应时间：68367.



2. 自定义一个动态网站，可以只含首页，然后用步进方式进行测试并见监看服务器性能。

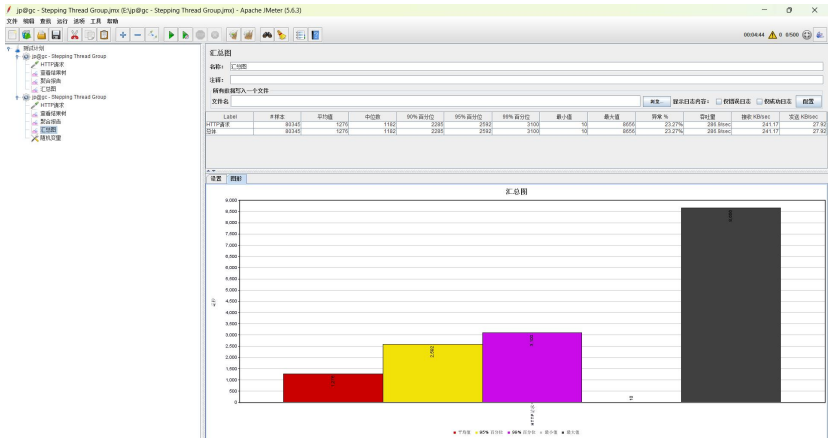
2.1 请求过程中服务器正确响应

```
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=40544 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=85111 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=78645 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=86948 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=69524 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=10864 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=52301 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=33863 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=32554 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=11509 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=86466 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=60502 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=65493 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=12625 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=93174 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=38587 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=96956 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=86596 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=14075 HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 00:55:47] "GET /hello?Name=3162 HTTP/1.1" 200 -
```

2.2 汇总图：异常率为 23.27%

平均响应时间：1276

最大响应时间：8656.

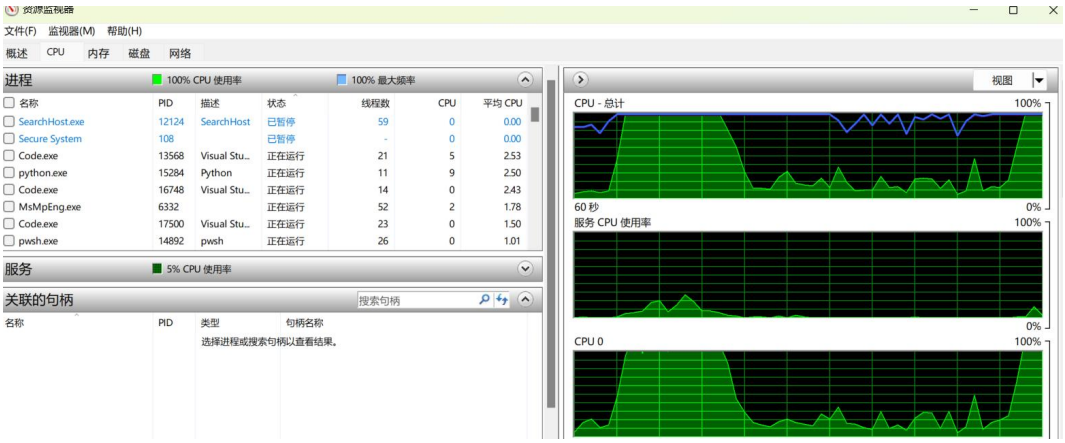


2.3 服务器性能

2.3.1 用户数为 200 以下

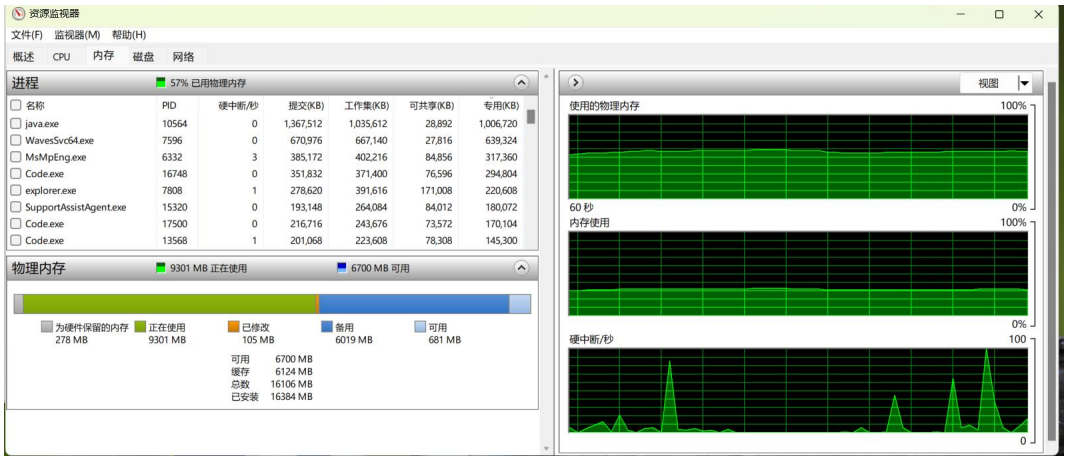
- **CPU 性能**

可以看见 CPU 利用率有低峰急剧上升至接近 100%.



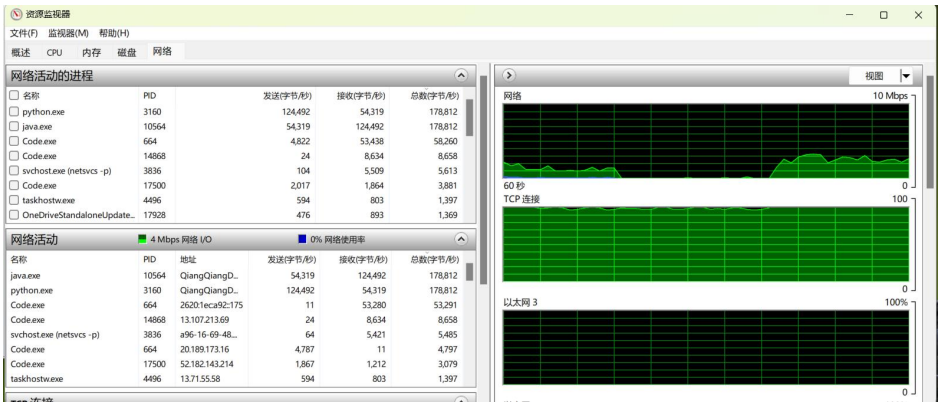
- **内存性能**

内存使用较为平稳，未发生较大波动，但硬中断偶尔出现峰值。



- **网络性能**

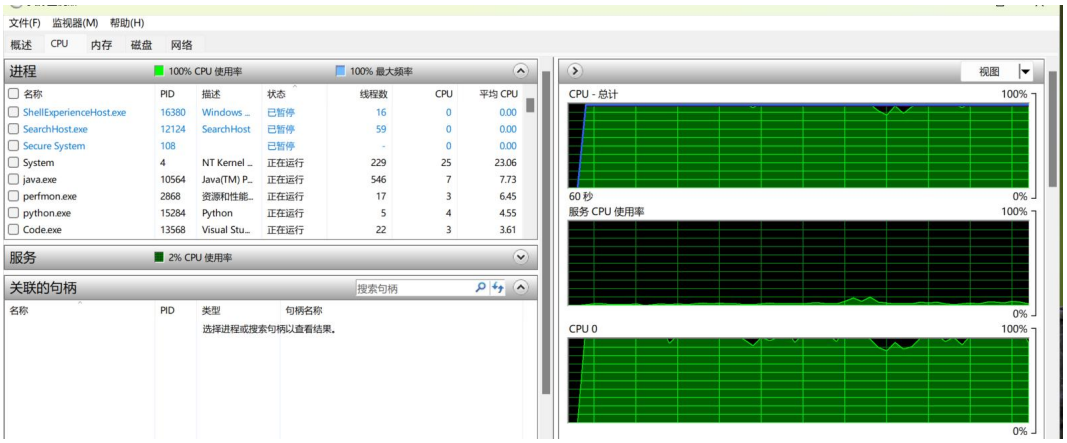
网络速率有所提高，TCP 连接保持最大值。



2.3.2 用户数为 500

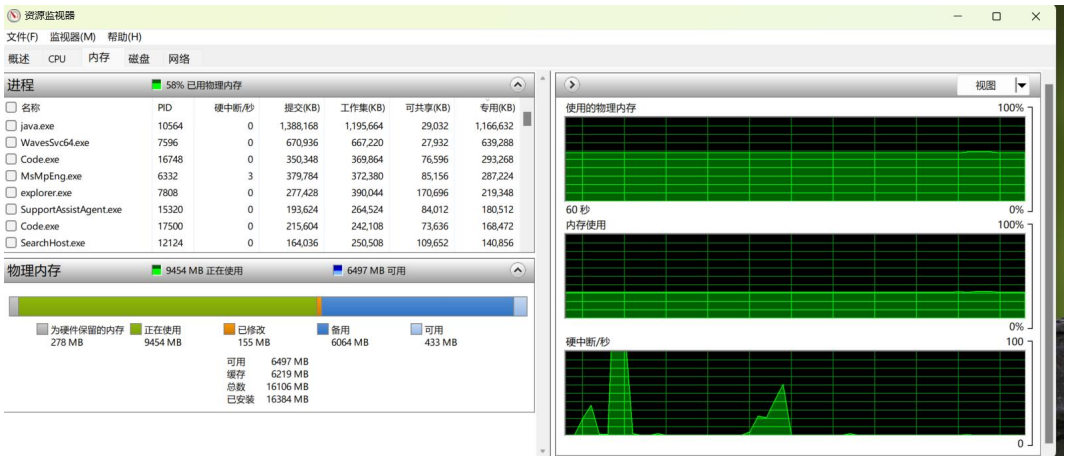
- **CPU 性能**

CPU 利用率长期维持在 100%左右。



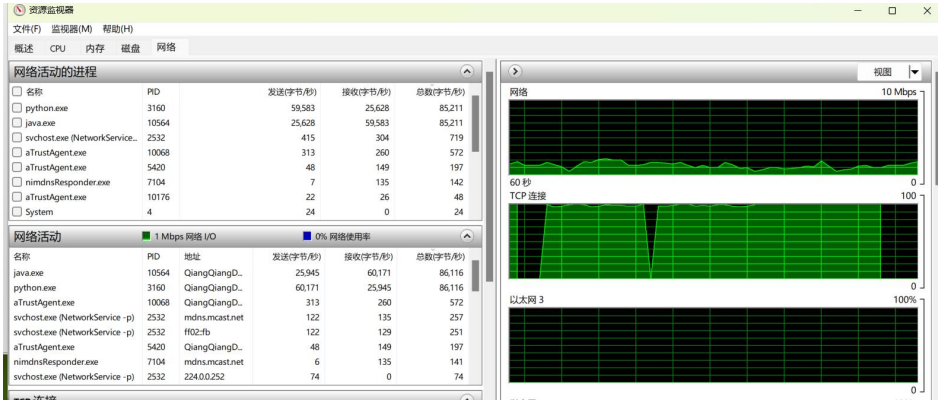
- **内存性能**

硬中断峰值有所减少，内存使用未发生较大波动。



- **网络性能**

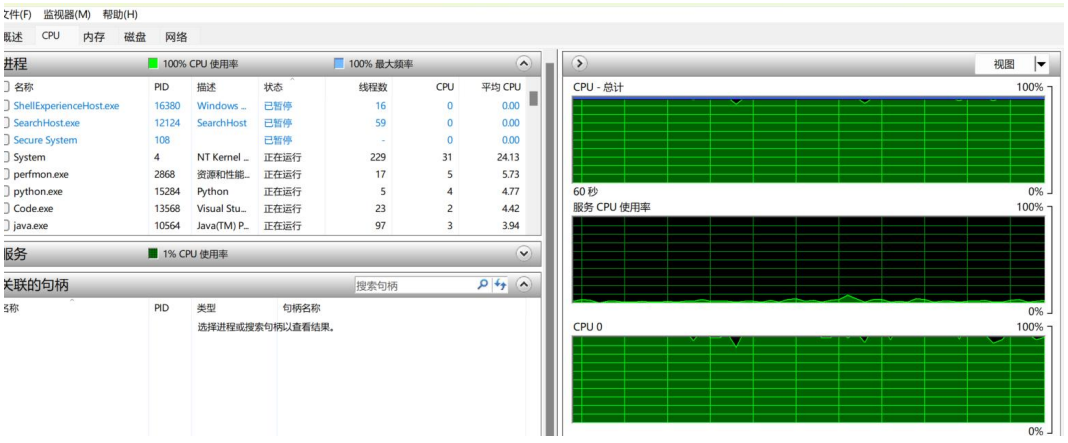
网络速率降低，TCP 连接长期处于峰值。



2.3.3 用户数下降到 50

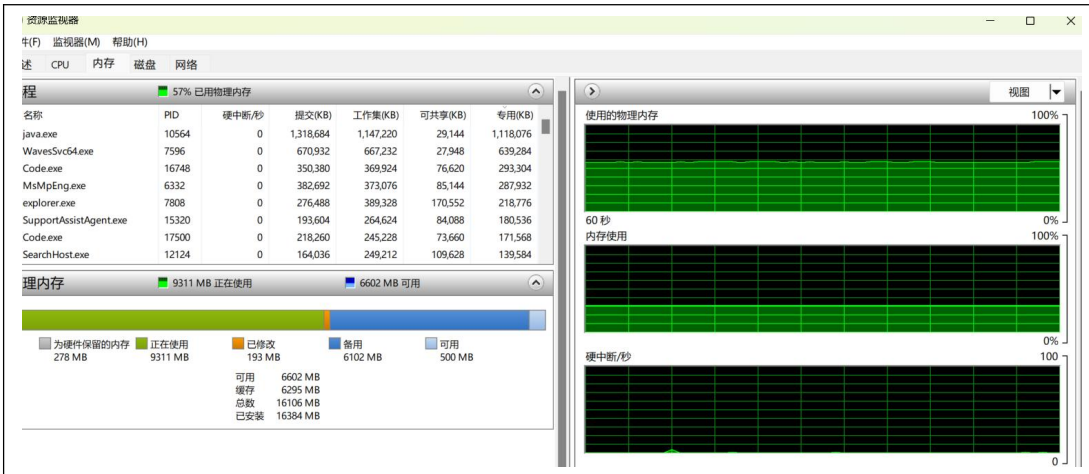
- **CPU 性能**

可以看见 CPU 利用率维持在 100%左右.



- **内存性能**

硬中断几乎为 0，内存使用未发生较大波动。



- **网络性能**

在整个过程结束后网络速率下降为 0，TCP 连接出现间隔峰值后变为 0.