

# 湖南大学

HUNAN UNIVERSITY



对对对队

201808010829 张继伟（组长）

201808010824 谢正宇（组员）

---

## 目录

### 实验二

一、 实验概要 .....	3
二、 性能测试 .....	6

# 实验二

## 一、实验概要

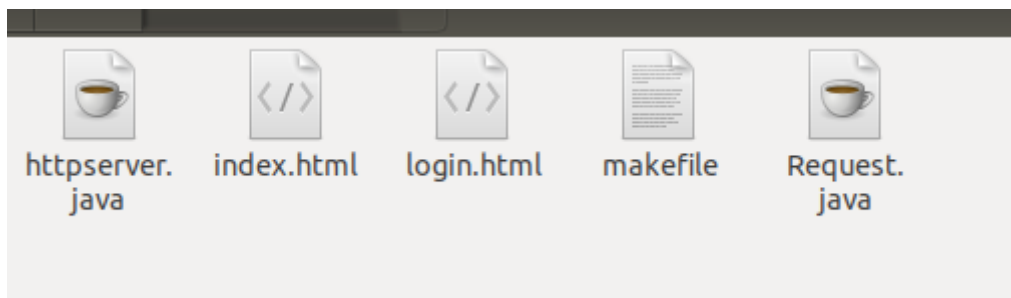
### 1. 实验环境

操作系统：ubuntu18.04

应用程序：apache

计算机语言：JAVA

### 2. 代码说明



本工程主要由两个类组成。

httpserver 类用于处理 http 请求并响应；

Request 类用于存储 http 请求信息；

### 3. 样例测试说明

进入 httpserver 文件夹执行命令：make clean

```
bash: export: " ": 不是有效的标识符
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$ make clean
rm *.class
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$
```

编译执行命令：make

```
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$ make
javac httpserver.java
javac Request.java
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$
```

运行程序（线程数为 x）执行命令：

```
java httpserver --ip 127.0.0.1 --port 8888 --number-thread xx(线程数)
```

```
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$ java httpserver --ip 127.0.0.1 --port 8888 --number-thread 100
```

再开一个终端进行 ab 压力测试（x 为请求数，y 为并发请求数）：

```
ab -n x -c y -k 127.0.0.1:8888/
```

```
zhangjiwei@zhangjiwei-virtual-machine:~$ ab -n 10000 -c 10 -k 127.0.0.1:8888/
This is ApacheBench, Version 2.3 <$Revision: 1807734 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests
```

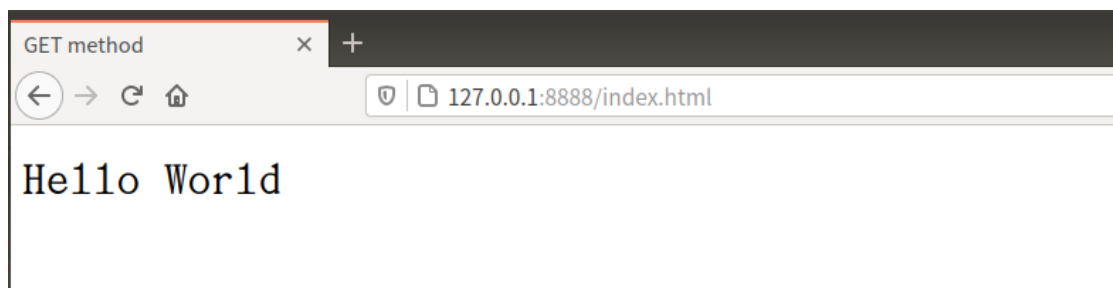
注意：切换线程数需要手动杀死前一个进程执行命令：`fuser -n tcp -k 8888`

```
[1]+ 已停止                  java httpserver --ip 127.0.0.1 --port 8888 --number-thread 100
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$ fuser -n tcp -k 8888
8888/tcp: 6217
[1]+ 已杀死                  java httpserver --ip 127.0.0.1 --port 8888 --number-thread 100
zhangjiwei@zhangjiwei-virtual-machine:~/httpserver$
```

## 4. 服务器测试

使用 GET 的请求

请求成功



```
curl -i -X GET http://127.0.0.1:8888/index.html
```

```
kizzy@kizzy-VirtualBox:~$ curl -i -X GET http://127.0.0.1:8888/index.html
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>GET method</title>
</head>
<body>
  <h1>Hello World</h1>
</body>
</html>
```

请求失败

```
kizzy@kizzy-VirtualBox:~$ curl -i -X GET http://127.0.0.1:8888/index.htm
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8

<h2>Not Found</h2>
```

```
kizzy@kizzy-VirtualBox:~$ curl -i -X GET http://127.0.0.1:8888/images/
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8

<h2>Not Found</h2>
```

使用 POST 的请求

请求成功

```
kizzy@kizzy-VirtualBox:~$ curl -i -X POST --data 'Name=HNU&ID=CS06142' http://127.0.0.1:8888/Post_show
HTTP/1.1 200 OK
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>POST method</title>
</head>
<body>
  Your Name:HNU
  ID:CS06142
</body>
</html>
```

请求失败

```
kizzy@kizzy-VirtualBox:~$ curl -i -X POST --data 'Name=HNU&ID=CS06142' http://127.0.0.1:8888/non_existence_post_show
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>POST method</title>
</head>
<body>
  <h1>Not Found</h1>
</body>
</html>
```

```
kizzy@kizzy-VirtualBox:~$ curl -i -X POST --data 'Incompatible_key1=val1&Incompatible_key2=val2' http://127.0.0.1:8888/Post_show
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>POST method</title>
</head>
<body>
  <h1>Not Found</h1>
</body>
</html>
```

## 二、性能测试

请求数对性能的影响

线程个数	请求数	并发数	Requests per second	Time taken for tests
1	10	10	94	0.106s
1	100	10	1012	0.099s
1	1000	10	2025	0.494s
1	10000	10	3628	2.756s
1	100000	10	6420	15.575s
1	1000000	10	6728	148.620s
100	10	10	204	0.049s

线程个数	请求数	并发数	Requests per second	Time taken for tests
100	100	10	669	0.149s
100	1000	10	2245	0.445s
100	10000	10	4568	2.189s
100	100000	10	5308	18.837s
100	1000000	10	6272	159.437s

由表格数据可以看出：无论单线程还是多线程，在并发请求数一定的情况下，随着请求数的增加，每秒处理的请求数随之增加，但是明显看出当请求数达到一定数量时，每秒处理的请求数增加缓慢；由于请求数呈倍数增长，所以每次测试的总时间也随之增加；再比较多线程和单线程的性能可以发现，单线程性能优于多线程，主要原因在于我们的虚拟机只分配了一个 CPU 核，所以在使用多线程时 I/O 操作过多花费了额外的时间。

#### 线程个数对性能的影响

线程个数	请求数	并发数	Requests per second	Time taken for tests
1	10000	10	3207	3.117s
10	10000	10	3417	2.926s
20	10000	10	3420	2.923s
40	10000	10	3710	2.695s
60	10000	10	3915	2.533s
80	10000	10	4307	2.257s
100	10000	10	4903	2.039s
1000	10000	10	4957	2.026s

由表格数据可以看出：在并发请求数和请求数一定的情况下，随着线程数量的增加，每秒处理的请求数随之增加，每次测试所需时间随之减少；分析原因在于线程数目未饱和之前，线程数目越多，可处理请求越多，性能越好。

#### 并发请求数对性能的影响

线程个数	请求数	并发数	Requests per second	Time taken for tests
100	1000	1	4595	0.218
100	1000	10	5093	0.193s
100	1000	20	5086	0.197s
100	1000	30	5006	0.200s
100	1000	40	5145	0.194s
100	1000	50	4996	0.200s
100	1000	60	5047	0.198s
100	1000	70	4945	0.202s
100	1000	80	5015	0.199s
100	1000	90	5105	0.196s

由表格数据可以看出：在线程数和请求数一定的情况下，随着并发请求数的增加，每秒处理的请求数随之增加，每次测试所需时间随之减少，分析发现原因在于我们设置的线程个数为 100，而控制变量逐次增加并发请求数，但是并发请求数均小于设置的线程数。