Lab2 test report

1. 实验概要

超文本传输协议（HTTP）是今天上网。与许多网络协议一样，HTTP 使用客户机-服务器模型。HTTP 客户端打开到 HTTP 服务器的网络连接并发送 HTTP 请求消息。那么服务器用一个 HTTP 响应消息进行响应，该消息通常包含一些资源（文件，文本和，客户端请求的。在这个实验中需要实现两方面的性能测试：需要测试在不同的服务器上运行时，服务器每秒可以处理多少 HTTP 请求；利用 ab 测试方法运行在同一台计算机上有多个客户端程序来模拟多个客户端。

1.1 程序输入

程序将在控制台接收用户输入，使用./httpserver --ip 127.0.0.1 --port 8888 --number-thread 8 命令来开启一个服务器，接下来输入 curl -i -X GET http://127.0.0.1:8888/index.html 来获取信息

1.2 程序输出

实验中结果输出到控制台。

(1) 使用 GET 方法

```
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X GET 127.0.0.1:8888/index.
html
HTTP/1.1 200 OK
Server: Sword'Server
Content-Length: 297
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//IETF//DTO HTML 2.0//EN">
<html><head>
<title>CS06142</title>
</head><body>
<h1 align="center">CS06142</h1>
<p align="center">Welcome to Cloud Computing Course.<br />
</p>
<hr>
<address align="center">Http Server at ip-127.0.0.1 Port 8888</address></body></
html>
```

```
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X GET 127.0.0.1:8888/Guolab
/non_existent.html
HTTP/1.1 404 Not Found
Server: Swordmen's Server
Content-Type:text/html
Content-length:114

<html><title>404 Not Found</title><body bgcolor=ffffff
 Not Found
<hr><em>HTTP Web server</em>
</body></html>
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X GET 127.0.0.1:8888/Empty_
dir/
HTTP/1.1 404 Not Found
Server: Swordmen's Server
Content-Type:text/html
Content-length:114

<html><title>404 Not Found</title><body bgcolor=ffffff
 Not Found
<hr><em>HTTP Web server</em>
</body></html>
```

(2) 使用 POST 方法

```
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X POST --data 'Name=HNU&ID=
CS06142' http://127.0.0.1:8888/Post_show
HTTP/1.1 200 OK
Server:Swordmen's Server
Content-Type: text/html
Content-length: 137

<html><title>Post method</title></head><body bgcolor=ffffff>
Your Name: HNU
ID: CS06142
<hr><em>Http Web server</em>
</body></html>
```

```
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X POST --data 'Name=HNU&ID=
CS06142' 127.0.0.1:8888/non_existent_post_show
HTTP/1.1 404 Not Found
Server: Swordmen's Server
Content-Type:text/html
Content-length:114

<html><title>404 Not Found</title><body bgcolor=ffffff
 Not Found
<hr><em>HTTP Web server</em>
</body></html>
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X POST --data 'Incompatible
_key1=val1&Incompatible_key2=val2' 127.0.0.1:8888/Post_show
HTTP/1.1 404 Not Found
Server: Swordmen's Server
Content-Type:text/html
Content-length:114

<html><title>404 Not Found</title><body bgcolor=ffffff
 Not Found
<hr><em>HTTP Web server</em>
</body></html>
```

(3) 其他方法

```
hxf@ubuntu:~/Desktop/httpserver/httpserver$ curl -i -X DELETE http://127.0.0.1:8
888/index.html
HTTP/1.1 501 Not Implemented
Server: Swordmen's Server
Content-Type:text/html
Content-length: 183

<html><title>501 Not Implemented</title></head><body bgcolor=ffffff>
Not Implemented
<p>Does not implement this method: DELETHTTP/1.1
<hr><em>Http Web server</em>
</body></html>
```

1.3 性能

使用 ab 测试，测试方法如下：ab -n 1000 -c 10
http://127.0.0.1:8888/index.html
Apache 的 ab 命令模拟多线程并发请求，测试服务器负载压力，也可以测试 nginx、lighthttp、IIS 等其它 Web 服务器的压力。
ab 命令对发出负载的计算机要求很低，既不会占用很多 CPU，也不会占用太多的内存，但却会给目标服务器造成巨大的负载，因此是某些 DDOS 攻击之必备良药，老少皆宜。自己使用也须谨慎。否则一次上太多的负载，造成目标服务器直接因内存耗光死机，而不得不硬重启，得不偿失。
在带宽不足的情况下，最好是本机进行测试，建议使用内网的另一台或者多台服务器通过内网进行测试，这样得出的数据，准确度会高很多。远程对 web 服务器进行压力测试，往往效果不理想（因为网络延时过大或带宽不足）
ab -n 800 -c 800 http://192.168.0.10/ 5 （-n 发出 800 个请求，-c 模拟 800 并发，相当 800 人同时访问，后面是测试 url）6 ab -t 60 -c 100 http://192.168.0.10/ 7 在 60 秒内发请求，一次 100 个请求。

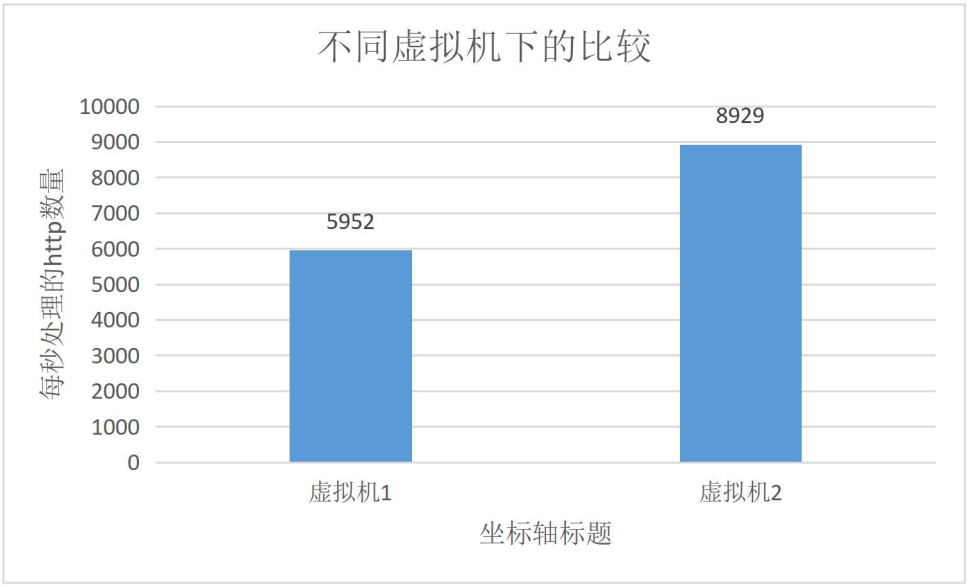
2. 性能测试

2.1 测试在不同的服务器上运行时，服务器每秒可以处理多少 http 请求
（1）不同虚拟机

```
Concurrency Level:      10
Time taken for tests:   0.168 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    5935.49 [#/sec] (mean)
Time per request:       1.685 [ms] (mean)
Time per request:       0.168 [ms] (mean, across all concurrent requests)
Transfer rate:          2225.81 [Kbytes/sec] received
```

```
Concurrency Level:      10
Time taken for tests:   0.112 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    8967.56 [#/sec] (mean)
Time per request:       1.115 [ms] (mean)
Time per request:       0.112 [ms] (mean, across all concurrent requests)
Transfer rate:          3362.84 [Kbytes/sec] received
```

不同的虚拟机版本下性能会有差异，新版本的性能更好，但不排除可能数由于网络等原因造成的差异

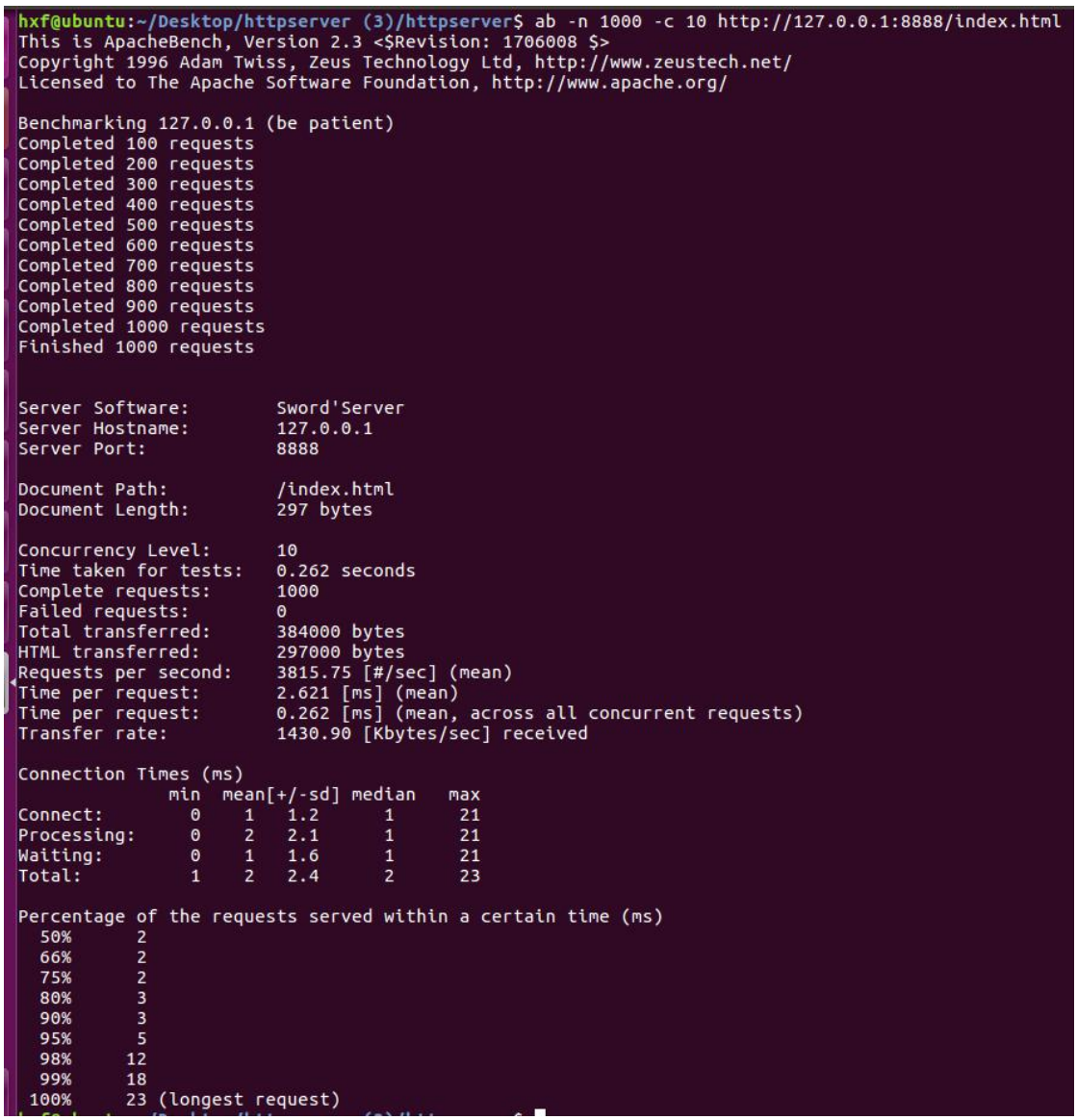


（2）不同内存大小

一共发送了 1000 个请求并发数为 10

(a) 在 2G 内存下运行结果如下：

我们可以看出来处理 1000 个 http 请求花费了 0.2622seconds，可知此时每秒可以处理大约 3814 个 http 请求

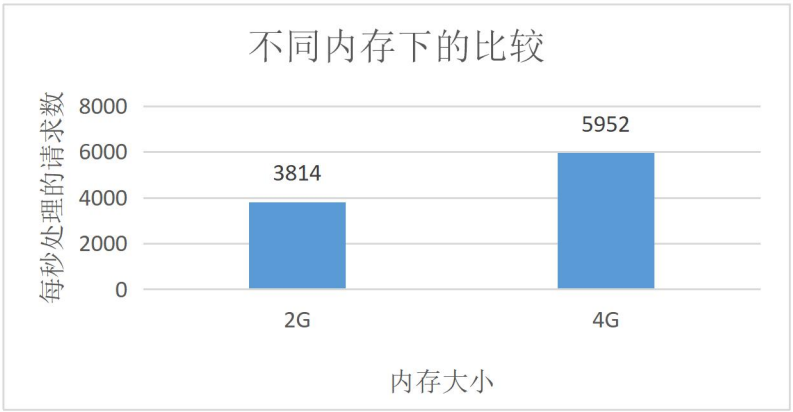


(b) 在 4G 下运行的结果如下：

我们可以看出来处理 1000 个 http 请求花费了 0.168seconds，可知此时每秒可以处理大约 5952 个 http 请求

```
hxf@ubuntu:~/Desktop/httpserver (3)/httpserver$ ab -n 1000 -c 10 http://127.0.0.1:8888/index.html
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
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 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Sword'Server
Server Hostname:        127.0.0.1
Server Port:            8888

Document Path:          /index.html
Document Length:        297 bytes

Concurrency Level:      10
Time taken for tests:   0.168 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    5935.49 [#/sec] (mean)
Time per request:       1.685 [ms] (mean)
Time per request:       0.168 [ms] (mean, across all concurrent requests)
Transfer rate:          2225.81 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.3      0       3
Processing:     0    1   2.3      1      19
Waiting:        0    1   2.2      1      18
Total:          0    2   2.3      1      19

Percentage of the requests served within a certain time (ms)
  50%      1
  66%      1
  75%      2
  80%      2
  90%      2
  95%      2
  98%     15
  99%     18
 100%     19 (longest request)
hxf@ubuntu:~/Desktop/httpserver (3)/httpserver$
```

(c) 结果比较



通过比较可知，在大内存上面运行相同的请求和并发数，运行时间会较短，这样相同条件下

每秒处理的 http 请求数会多一些

2.2 不同数量的并发客户端

并发客户端为 5：

```
Concurrency Level:      5
Time taken for tests:   0.135 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    7388.63 [#/sec] (mean)
Time per request:       0.677 [ms] (mean)
Time per request:       0.135 [ms] (mean, across all concurrent requests)
Transfer rate:          2770.74 [Kbytes/sec] received
```

并发客户端为 6：

```
Concurrency Level:      6
Time taken for tests:   0.147 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    6818.17 [#/sec] (mean)
Time per request:       0.880 [ms] (mean)
Time per request:       0.147 [ms] (mean, across all concurrent requests)
Transfer rate:          2556.81 [Kbytes/sec] received
```

并发客户端为 7：

```
Concurrency Level:      7
Time taken for tests:   0.123 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    8121.37 [#/sec] (mean)
Time per request:       0.862 [ms] (mean)
Time per request:       0.123 [ms] (mean, across all concurrent requests)
Transfer rate:          3045.51 [Kbytes/sec] received
```

并发客户端为 8：

```
Concurrency Level:      8
Time taken for tests:   0.167 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    5976.43 [#/sec] (mean)
Time per request:       1.339 [ms] (mean)
Time per request:       0.167 [ms] (mean, across all concurrent requests)
Transfer rate:          2241.16 [Kbytes/sec] received
```

并发客户端为 9：

```
Concurrency Level:      9
Time taken for tests:   0.061 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    16357.24 [#/sec] (mean)
Time per request:       0.550 [ms] (mean)
Time per request:       0.061 [ms] (mean, across all concurrent requests)
Transfer rate:          6133.97 [Kbytes/sec] received
```

并发客户端为 10：

```
Concurrency Level:      10
Time taken for tests:   0.112 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    8967.56 [#/sec] (mean)
Time per request:       1.115 [ms] (mean)
Time per request:       0.112 [ms] (mean, across all concurrent requests)
Transfer rate:          3362.84 [Kbytes/sec] received
```

并发客户端为 11：

```
Concurrency Level:      11
Time taken for tests:   0.151 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    6623.79 [#/sec] (mean)
Time per request:       1.661 [ms] (mean)
Time per request:       0.151 [ms] (mean, across all concurrent requests)
Transfer rate:          2483.92 [Kbytes/sec] received
```

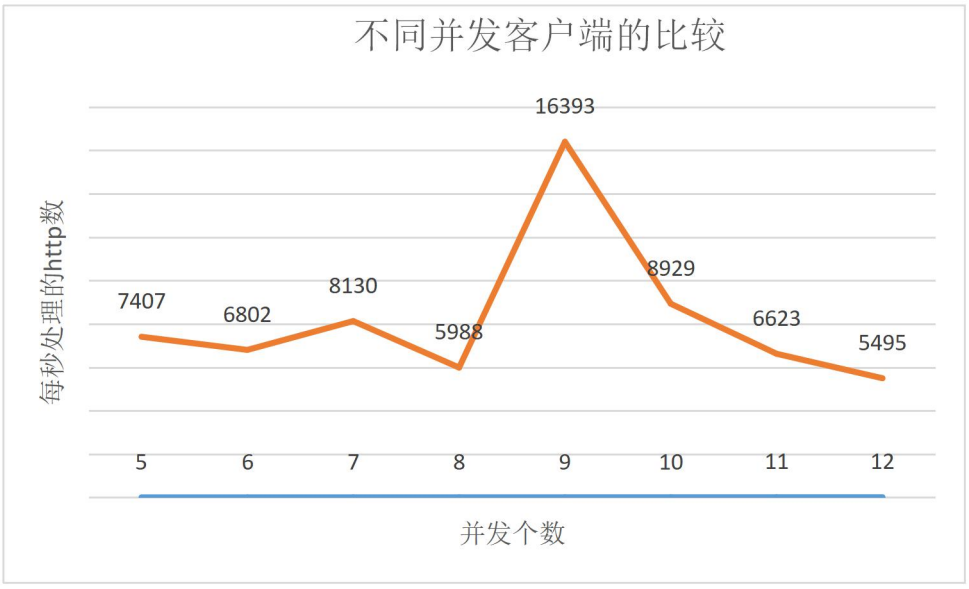
并发客户端为 12：

```
Concurrency Level:      12
Time taken for tests:   0.182 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      384000 bytes
HTML transferred:       297000 bytes
Requests per second:    5489.95 [#/sec] (mean)
Time per request:       2.186 [ms] (mean)
Time per request:       0.182 [ms] (mean, across all concurrent requests)
Transfer rate:          2058.73 [Kbytes/sec] received
```

进行数据处理得到每秒处理的 http 请求数

5	7407
6	6802
7	8130
8	5988
9	16393
10	8929
11	6623
12	5495

比较结果



可以看出在并发数为 9 的时候性能最好，这是因为在这点的线程使用效率最高

三、总结

在不同的环境下运行测试的效果会有差异，总的来说内存大的时候每秒处理的 http 请求数量多，新版本的虚拟机可能运行效果更好。同时我们改变并发的客户端的数量，发现在并发数为 9 的时候每秒处理的 http 请求数最多，达到了惊人的 16393 个。