## 1: 安装和介绍

### 1.1介绍：

**ApacheBench 是 Apache 服务器自带的一个web压力测试工具，简称ab。ab又是一个命令行工具，对发起负载的本机要求很低，根据ab命令可以创建很多的并发访问线程，模拟多个访问者同时对某一URL地址进行访问，因此可以用来测试目标服务器的负载压力**

### 1.2下载安装：

#### 1.2.1：下载地址：<https://httpd.apache.org/download.cgi>

#### 1.2.2：Windows64位下载地址：

<https://www.apachelounge.com/download/VC15/binaries/httpd-2.4.29-Win64-VC15.zip>

### 1.3官方文档地址:相关参数说明:

**1.3.1：https://httpd.apache.org/docs/2.4/programs/ab.html**

## 2:简单介绍

#### 2.1：环境配置

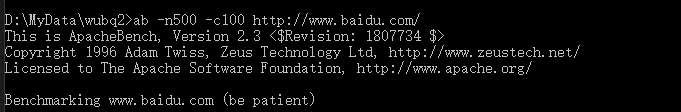
**将D:\wubiqin\util\apache bench\Apache24\bin配置到环境变量中**

**这样就可以直接在命令行中直接使用ab进行测试**

#### 2.2：简单并发压测百度（发出500次请求，并发级别为100，相当于100人同时访问百度）

**访问规范:（端口没有输入 ab默认为80 abs默认为443）**

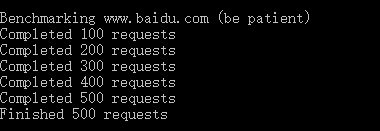


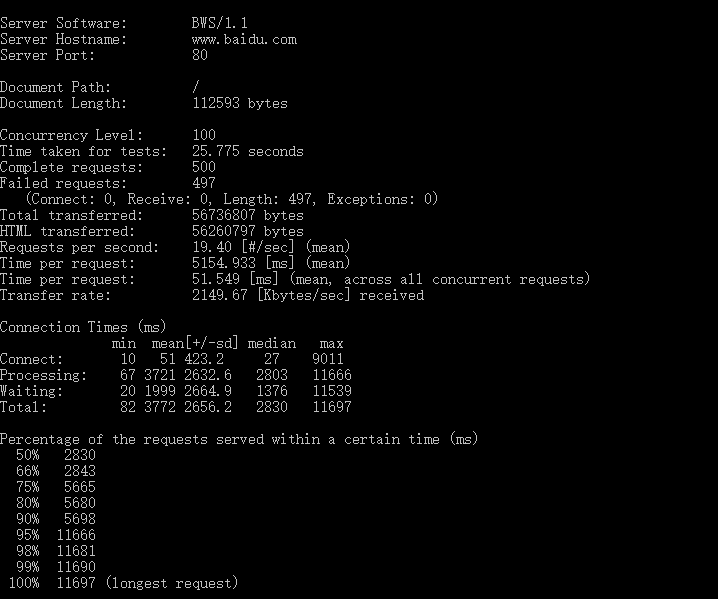


##### 2.2.1：Cmd中输入命令：ab -n500 -c100 <http://www.baidu.com/>

##### 2.2.2：返回的参数展示：

###### 2.2.2.1发出请求





## 3：主要参数介绍(默认get请求)

**-n : 这次测试请求的次数（默认执行一次请求）**

**-c : 一次产生请求的个数（默认一个）**

**-t : 测试进行的时间（单位为秒，内部隐含值 –n50000，默认没有时间限制）**

**-p : 发送post请求 （-p 与-T要结合使用）**

**发送post请求（使用方法：-p 文件地址 –T content\_type**

**信息）**

**-u ： 发送put请求（-p 与-T要结合使用）**

**发送put请求（使用方法：-u 文件地址 –T content\_type**

**信息）**

**-H : Header信息（例子：-H "Token=01c329aef1-1741-464c-8014-5e6fa6e4a98c"）**

**可增加多个（例子：-H “a：1” –H“b：2”）**

**-C : Cookie值（例子-C "sid=01c329aef1-1741-464c-8014-5e6fa6e4a98c"）**

**可增加多个 （例子：例子：-C “a：1” –C“b：2”）**

## 4:返回参数:介绍

**Server Software: BWS/1.1**

**Server Hostname: www.baidu.com**

**Server Port: 80**

**Document Path: /**

**Document Length: 112593 bytes 页面大小**

**Concurrency Level: 100 测试的并发数**

**Time taken for tests: 25.775 seconds 请求处理完成消耗时间**

**Complete requests: 500 完成的请求数**

**Failed requests: 497 失败的请求数**

**(Connect: 0, Receive: 0, Length: 497, Exceptions: 0)**

**Total transferred: 56736807 bytes 网络传输总量**

**HTML transferred: 56260797 bytes html 内容传输量**

**Requests per second: 19.40 [#/sec] (mean) 每秒请求数 吞吐量（每秒事务数）**

**Time per request: 5154.933 [ms] (mean) 一次并发请求响应的时间（平均事务响应时间）**

**Time per request: 51.549 [ms] (mean, across all concurrent requests) 并发时 每次请求响应的时间**

**Transfer rate: 2149.67 [Kbytes/sec] received 平均每秒网络上的流量= Total transferred /Time taken for tests**

**网络上消耗的时间分解：**

**Connection Times (ms)**

**min mean[+/-sd] median max**

**Connect: 10 51 423.2 27 9011**

**Processing: 67 3721 2632.6 2803 11666**

**Waiting: 20 1999 2664.9 1376 11539**

**Total: 82 3772 2656.2 2830 11697**

**整个场景中所有请求的响应情况**

**Percentage of the requests served within a certain time (ms)**

**50% 2830**

**66% 2843**

**75% 5665**

**80% 5680**

**90% 5698**

**95% 11666**

**98% 11681**

**99% 11690**

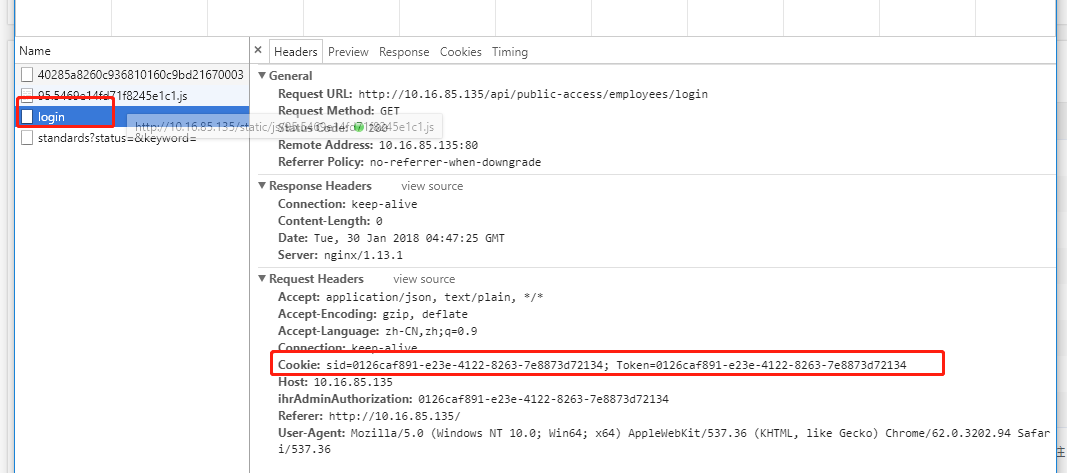
**100% 11697 (longest request)**

**50%的响应时间小于2830毫秒，66%的响应时间小于2843毫秒，75%的响应时间小于5665毫秒，80%的响应时间小于5680毫秒，90%的响应时间小于5698毫秒，95%的响应时间小于11666毫秒，98%的响应时间小于11681毫秒，99%的响应时间小于11690毫秒，最大响应时间小于11697**

## 4:示例

**首先登录系统：**

**抓包获得Cookie**



### 4.1：get请求

#### 4.1.1：Cmd中输入命令:

#### ab –n500 –c100 -C “Token=01704483ad-e6d7-4f16-ba1c-444ee3562061” <http://10.16.85.135/#/compensation/base-setting/policy-requirements/national-tax-rate>



### 4.2：Post请求

#### 4.2.1：Cmd中输入命令：

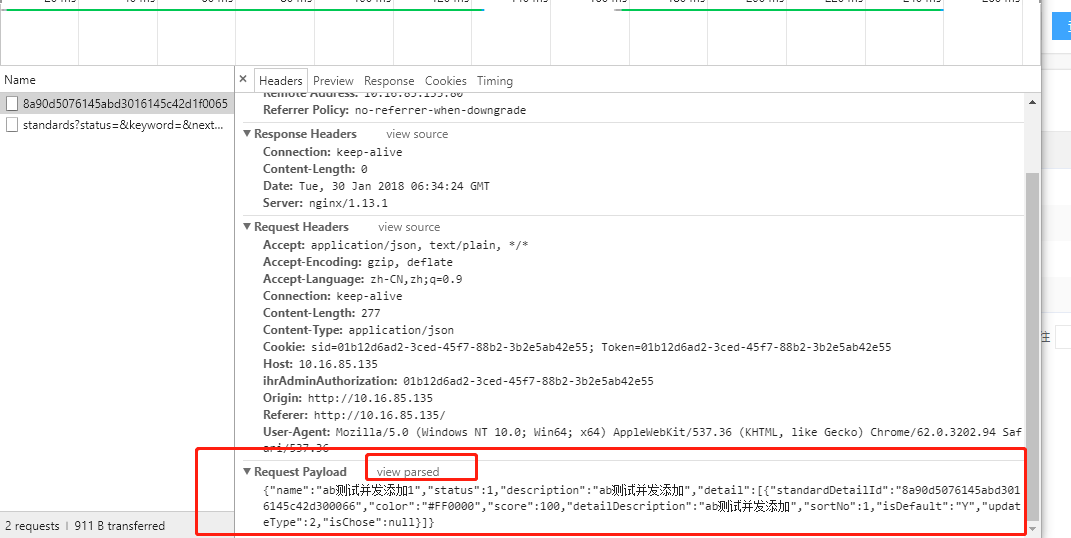
ab –n100 –c50 –p D:\MyData\wubq2\IdeaProjects\springbootstarterhello\src\main\resources\data.txt –T application/json –C ”sid=01b12d6ad2-3ced-45f7-88b2-3b2e5ab42e55” <http://10.16.85.135/api/grape/standard>

（-p 文件地址）:文件中的内容（好像要在编译器中生成txt例如在idea，elipse）

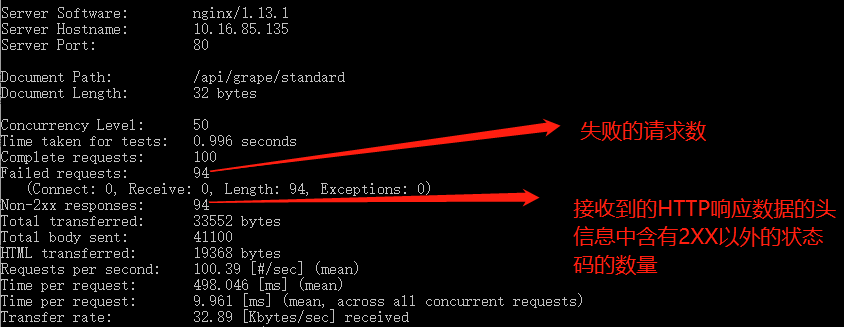
#### 4.2.2：Data.txt

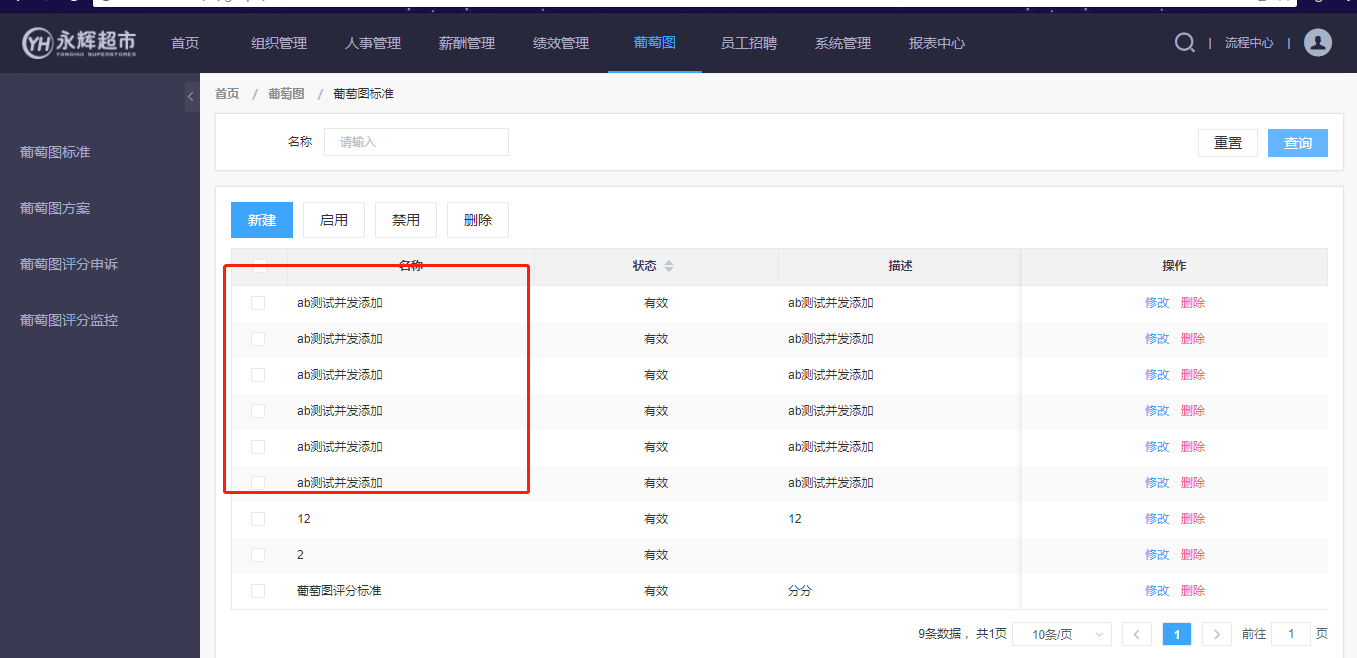
{"name":"ab测试并发添加","detail":[{"sortNo":1,"updateType":1,"score":100,"color":"#FF0000","isDefault":"Y","detailDescription":"ab测试并发添加"}],"status":1,"description":"ab测试并发添加"}

#### 4.2.3:Data.txt中的内容通过抓包获得



#### 4.2.4请求结果





### 4.3：Put请求:

#### 4.3.1:Cmd中输入命令：

ab -u D:\MyData\wubq2\IdeaProjects\springbootstarterhello\src\main\resources\data.txt -T application/json -C "sid=01b12d6ad2-3ced-45f7-88b2-3b2e5ab42e55" <http://10.16.85.135/api/grape/standards/8a90d5076145abd3016145c42d1f0065>

#### 4.3.2:data.txt

{"name":"ab测试并发添加1","status":1,"description":"ab测试并发添加1","detail":[{"standardDetailId":"8a90d5076145abd3016145c42d300066","color":"#FF0000","score":100,"detailDescription":"ab测试并发添加","sortNo":1,"isDefault":"Y","updateType":2,"isChose":null}]}

#### 4.3.3:返回结果：

