需求：

开发一个 http 性能压测工具，压测 nginx 的性能，类似 ab，具体要求如下：

1. C 语言开发。

2. 程序支持以下参数：线程数、总连接数、并发数。

3. 输出的结果为每秒完成的 http 连接数。

输入参数如下：

-c 并发连接数目

-t 设置的线程数目

-s 总的连接数目

-r 程序的超时时间 （默认为 30s）

-9 使用 Http/0.9 测试包进行压测

-1 使用 Http/1.0 测试包进行压测（默认为 1.0）

-2 使用 Http/1.1 测试包进行压测

测试用例：

1. 本地局域网用例

本地测试环境为虚拟机内 i7-4710MQ 双核心处理器，3GB 内存，10MB/s 网络，局域网内传输 200kB 静态页面。

结果如图所示。

```
zhy@ubuntu:~/SerTanaBenchmark$ /home/zhy/SerTanaBenchmark/SerTanaBenchmark -c 10000 -t 10 -s 50000 -r 10 http://127.0.0.1:8000
------------------------------
SerTanaWebBench
Trying to connect: GET http://127.0.0.1:8000
10 numthreads are created for connecting 10000 concurrent connections
50000 total connections are trying to connect

Total connections are 50002, receive 9215380 bytes
Successful connection number rate is 24247/sec
Total requests: 48495 susceed, 1507 failed, and 2 sec is used to complete the task
------------------------------
```

测试本地局域网内 Http 站点，并发连接数为(-c 10000)，线程数为（-t 10），总连接数为（-s 50000），超时时间(-r 10 秒)

测试的结果：

工具发起的连接总数为 50002，其中成功建立了 48495 个连接，失败了 1507 例，程序运行时间为 2 秒，每秒成功连接率为 24274，整个过程接收了服务端的 9215380 字节的数据。

2. 公网用例

用例 2.1

```
zhy@ubuntu:~$ /home/zhy/SerTanaBenchmark/SerTanaBenchmark -c 8000 -t 10 -s 20000
 -r 30 http://www.qq.com
------------------------------
SerTanaWebBench
Trying to connect: GET http://www.qq.com
10 numthreads are created for connecting 8000 concurrent connections
20000 total connections are trying to connect

Total connections are 11671, receive 255662372 bytes
Successful connection number rate is 388/sec
Total requests: 11668 susceed, 3 failed, and 30 sec is used to complete the task
------------------------------
```

测试 http://www.qq.com 站点，并发连接数为(-c 8000)，线程数为（-t 10），总连接数为（-s 20000），超时时间(-r 30 秒)

测试的结果：

工具发起的连接总数为 11671（程序运行超时），其中成功建立了 11668 个连接，失败了 3 例，程序运行时间为 30 秒，每秒成功连接率为 388，整个过程接收了服务端的 255662372 字节的数据。

总结：

1、传统的压测工具一般的实现是并发数直接等于开辟的进程数(webbench）或者线程数(Siege)，这样模拟并发请求的原理会更加直观，模拟效果在小并发情形下较为准确，但是这种方法在实现单客户端模拟大量并发的情形时效率较为低下，一方面单机开辟的进程数和线程数有限，另一方面大量的进程和线程的上下文切换使程序性能下降较快。

2、另外一种实现方案是指定固定的线程数目进行模拟并发，这种方法的优势在于一个线程可以处理多个套接字，即一个线程可以建立多个连接。同时也减小了 CPU 上下文切换次数。但另一方面，这种方案下的线程往往会阻塞在读写 IO 操作上，从而影响模拟并发的性能。所以不得不降低 Socket 的读写 IO 的阻塞时间，但是这样一来就会提高连接失败率，从而导致了因为测试机性能的不足使得压测数据的结果与事实出现了偏差。

3.高性能的实现方式为基于 I/O 复用的方法，每个子线程维护一个 socket 池，通过 socket 池不断并发读写服务端，从而实现高性能并发测试的功能。

其中，socket 池 = 并发量 / 线程数目。