

# 3Proxy 压力测试方法

Author: 张中华

**背景知识:** 3Proxy 作为一个代理工具,起着连接用户和实际服务器的桥梁作用,用户所有的数据将通过 3Proxy 再传输到实际服务器,因此 3Proxy 本身的性能关系到我们的代理线路是否能够承受多并发,高负载,因此对 3Proxy 有必要做压力测试,由于我们的服务主要应用于游戏代理,因此最直接的压测方式是找很多人一起玩游戏,然而这种压测方式,耗时耗人力,如果每新开一条线路就找一堆人做压测,成本将高到无法承受的地步,因此有必要将压测步骤程序化,开发或者使用程序做压测,下文描述的是一种基于 **wget** 命令模拟用户游戏行为的压测方法。

**方案分析:** 多用户玩游戏本质上是多个客户端同时向服务端发出请求,玩游戏的过程是客户端和服务端不断地进行数据的交换,该过程可以等价于不断从游戏服务器下载文件,因此我们可以通过下载文件的方式模拟多用户同时玩游戏,如使用浏览器同时下载 10 个文件,模拟 10 个用户同时玩游戏,通过控制下载文件的大小和带宽限速控制下载时间,但是使用浏览器建立多个下载任务存在以下问题:

问题 1: 同时下载 10 个文件需要手工建立 10 个下载任务,如果要模拟 100 个呢,手工建立 100 个任务明显是不现实的,因此有必要使用命令下载任务,并在脚本中控制并发数。

**解决办法:** 使用 **wget** 或 **curl** 命令实现文件下载。

问题 2: 有必要自己架设文件服务器,因为如果使用下载网站提供的文件,该网站一般都会有限流,特别是当大量请求都来自同一个 IP 地址时,这将导致很多任务速度极低或者请求被拒绝。

**解决办法:** 使用 **nginx** 架设文件服务器,由于 **nginx** 的高性能,有足够的处理能力处理大并发。

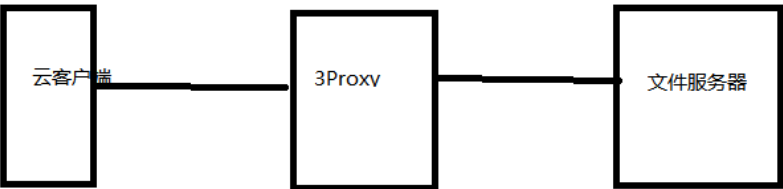
问题 3: 文件下载是通过代理服务器下载的,因此文件服务器不能离代理服务太远,否则代理服务器和文件服务器之间的距离将影响下载,导致测试结果异常,比如代理服务器本身具备足够的处理能力,然而文件服务器带宽小,距离远,那么文件服务器本身将成为线路的瓶颈,这样的话再怎么测,也测不出代理服务器的性能。

**解决办法:** 在代理服务器的附件假设文件服务器,如果代理服务器的出口在韩国,那么将文件服务器假设在韩国,同时文件服务器也需要具备和代理服务器匹配的硬件性能和带宽。

问题 4: 实际多个人玩游戏的时候,客户端是分散在多台电脑上的,如果压测也找一堆电脑模拟多个用户那显然不实际,很显然压测的客户端将集中在同一台主机上,因此压测的客户端需具备超强处理能力和和代理服务器对等的带宽。

解决办法：国内云上申请一台性能较强带宽足够的 windows 机器。

整个方案架构如下：



稍后将用Visio作图代替之

以下介绍 IP=42.157.196.49 3Proxy 服务器的压测方法，该机器归属地广东东莞  
步骤 1：申请一台国内的阿里云做文件服务器

实例ID/名称	标签	监控	可用区	IP地址	状态	网络类型	配置	付费方式
i-bp15o2hw454mcgk20u launch-advisor-20200811			华东 1 可用区 H	47.111.174.136(公网) 172.16.118.234 (私有)	运行中	专有网络	2 vCPU 8 GiB (I/O优化) ecs.hfg6.large 100Mbps (峰值)	按量 2020年8月11日 14:16 创建

步骤 2：在上面部署 Nginx 做文件服务器，nginx 参考配置如下：

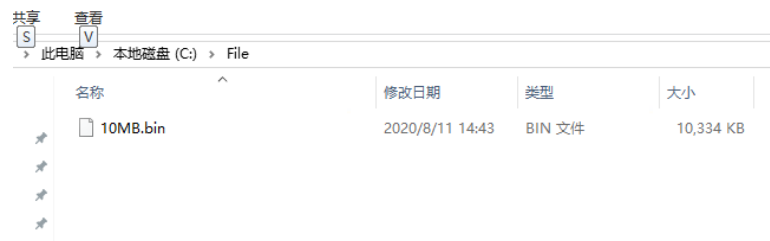
```
sendfile    on;
#tcp_nopush on;

#keepalive_timeout 0;
keepalive_timeout 65;

#gzip on;

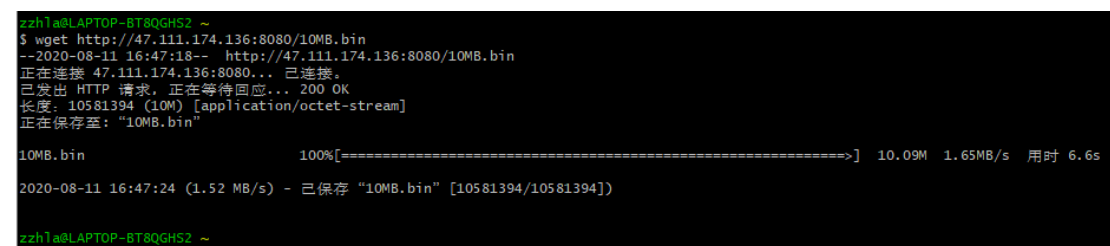
server {
    listen      8080;
    server_name localhost;
    charset gbk;
    root C:\File;

    location / {
        autoindex on;
        autoindex_exact_size off;
        autoindex_localtime on;
    }
}
```



测试文件目录

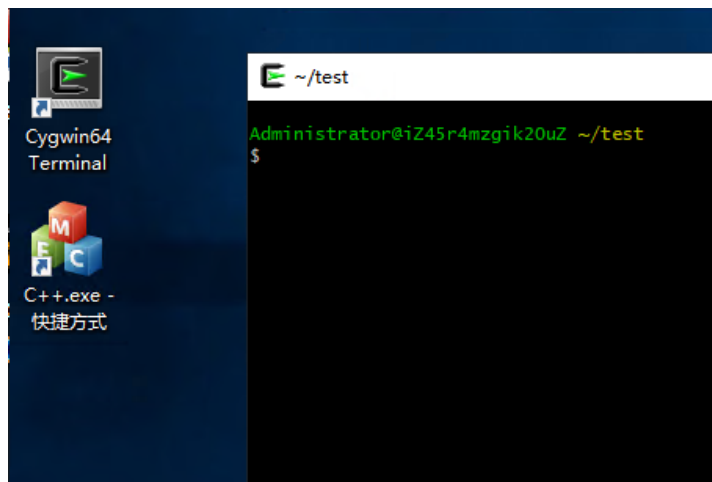
步骤三：使用浏览器查看文件浏览器并测试下载，或使用 wget 命令测试下载



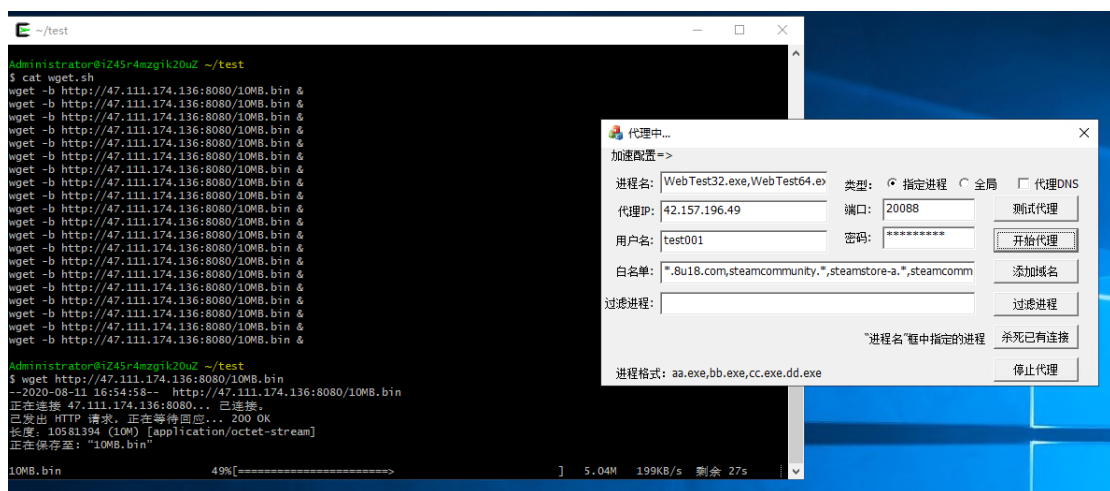
步骤四：申请一台性能较好，带宽足够的阿里云 windows 机器做测试客户端



步骤五：在 windows 机器上安装代理客户端和 cygwin，安装 cygwin 主要是因为 wget 的 windows 版似乎只在 cygwin 平台上维护，另外 cygwin 可模拟 linux 环境，方便编写 shell 脚本



步骤六：启动代理客户端，并设置代理进程 wget.exe，然后再 cygwin 里测试下载，观察下载速度和查看 3proxy 日志，确定 wget 代理成功



步骤七：编写脚本测试，测试并发，以下脚本为模拟 20 并发

```

Administrator@iZ45r4mzgik20uZ ~/test
$ cat wget.sh
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
wget -b http://47.111.174.136:8080/10MB.bin &
Administrator@iZ45r4mzgik20uZ ~/test
$

```

测试过程如下，下载过程都将被写入日志，稍后可查看日志获取下载速度，耗时能信息

```

Administrator@iZ45r4mzgik20uZ ~/test
$ ./wget.sh
继续在后台运行，pid 为 1909。
将把输出写入至 "wget-log.1"。
继续在后台运行，pid 为 1910。
继续在后台运行，pid 为 1911。
将把输出写入至 "wget-log"。
将把输出写入至 "wget-log.2"。
继续在后台运行，pid 为 1916。
将把输出写入至 "wget-log.5"。
继续在后台运行，pid 为 1917。
将把输出写入至 "wget-log.4"。
继续在后台运行，pid 为 1918。
将把输出写入至 "wget-log.6"。
继续在后台运行，pid 为 1913。
将把输出写入至 "wget-log.3"。
继续在后台运行，pid 为 1921。
将把输出写入至 "wget-log.7"。
继续在后台运行，pid 为 1923。
将把输出写入至 "wget-log.8"。
继续在后台运行，pid 为 1925。
将把输出写入至 "wget-log.9"。
继续在后台运行，pid 为 1927。
将把输出写入至 "wget-log.10"。
继续在后台运行，pid 为 1930。
继续在后台运行，pid 为 1931。
将把输出写入至 "wget-log.12"。
将把输出写入至 "wget-log.11"。
继续在后台运行，pid 为 1936。
继续在后台运行，pid 为 1937。
将把输出写入至 "wget-log.14"。
将把输出写入至 "wget-log.15"。
继续在后台运行，pid 为 1933。
将把输出写入至 "wget-log.13"。
继续在后台运行，pid 为 1940。
将把输出写入至 "wget-log.16"。

Administrator@iZ45r4mzgik20uZ ~/test
$ 继续在后台运行，pid 为 1941。
将把输出写入至 "wget-log.18"。
继续在后台运行，pid 为 1942。
将把输出写入至 "wget-log.17"。
继续在后台运行，pid 为 1943。
将把输出写入至 "wget-log.19"。
$

```

下载文件及日志

```
Administrator@iZ45r4mzgik20uZ ~/test
$ ll
总用量 186388
drwxr-xr-x+ 1 Administrator None          0 8月 11 16:57 .
drwxr-xr-x+ 1 Administrator None          0 8月 11 15:55 ..
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.1
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.1.1
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.1.2
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.1.3
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2.1
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2.2
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2.3
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2.4
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.2.5
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3.1
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3.2
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3.3
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3.4
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.3.5
-rw-r--r--  1 Administrator None 10581394 8月 11 14:43 10MB.bin.4
-rwxr-xr-x  1 Administrator None          920 8月 11 15:50 wget.sh
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log
-rw-r--r--  1 Administrator None 16233 8月 11 16:58 wget-log.1
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.10
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.11
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.12
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.13
-rw-r--r--  1 Administrator None 16233 8月 11 16:58 wget-log.14
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.15
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.16
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.17
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.18
-rw-r--r--  1 Administrator None 16233 8月 11 16:58 wget-log.19
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.2
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.3
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.4
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.5
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.6
-rw-r--r--  1 Administrator None 16233 8月 11 16:58 wget-log.7
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.8
-rw-r--r--  1 Administrator None 16237 8月 11 16:58 wget-log.9

Administrator@iZ45r4mzgik20uZ ~/test
$ |
```

步骤八：查看下载日志，以 wget-log.19 为例

```
$ cat wget-log.19
```

```
--2020-08-11 16:57:40-- http://47.111.174.136:8080/10MB.bin
```

正在连接 47.111.174.136:8080... 已连接。

已发出 HTTP 请求，正在等待回应... 200 OK

长度：10581394 (10M) [application/octet-stream]

正在保存至：“10MB.bin.4”

```
0K ..... 0% 199K 52s
50K ..... 0% 208K 50s
100K ..... 1% 208K 50s
150K ..... 1% 208K 49s
200K ..... 2% 208K 49s
250K ..... 2% 206K 49s
300K ..... 3% 209K 48s
```

350K	.....	3%	156K	50s
400K	.....	4%	208K	50s
450K	.....	4%	209K	49s
500K	.....	5%	208K	49s
550K	.....	5%	209K	48s
600K	.....	6%	208K	48s
650K	.....	6%	208K	48s
700K	.....	7%	208K	47s
750K	.....	7%	156K	48s
800K	.....	8%	208K	47s
850K	.....	8%	209K	47s
900K	.....	9%	208K	47s
950K	.....	9%	208K	46s

中间省略

1000K	.....	10%	209K	46s
1050K	.....	10%	208K	46s
1100K	.....	11%	208K	45s
1150K	.....	11%	156K	46s
1200K	.....	12%	208K	45s
1250K	.....	12%	209K	45s
9950K	.....	96%	156K	2s
10000K	.....	97%	207K	1s
10050K	.....	97%	208K	1s
10100K	.....	98%	208K	1s
10150K	.....	98%	207K	1s
10200K	.....	99%	195K	0s
10250K	.....	99%	208K	0s
10300K	.....	100%	208K=52s	

2020-08-11 16:58:32 (200 KB/s) - 已保存 “10MB.bin.4” [10581394/10581394])

从上面日志可看成，整个下载过程耗时 52s，速度稳定在 200K 左右，符合预期，为测试性能峰值，加大并发继续测试。