**Linux下安装Webbench工具进行压力测试**

**目录**  [隐藏](http://www.884358.com/webbench-test/)

[1 WebBench安装](http://www.884358.com/webbench-test/#WebBench_an_zhuang)

[2 WebBench使用](http://www.884358.com/webbench-test/#WebBench_shi_yong)

[3 控制并发数及请求频率](http://www.884358.com/webbench-test/#kong_zhi_bing_fa_shu_ji_qing_qiu_pin_lu)

[4 安装fail2ban](http://www.884358.com/webbench-test/#an_zhuangfail2ban)

[4.1 检查Firewalld是否启用](http://www.884358.com/webbench-test/#jian_chaFirewalld_shi_fou_qi_yong)

[5 配置规则](http://www.884358.com/webbench-test/#pei_zhi_gui_ze)

[6 监控nginx访问日志](http://www.884358.com/webbench-test/#jian_kongnginx_fang_wen_ri_zhi)

[7 常用命令](http://www.884358.com/webbench-test/#chang_yong_ming_ling)

[8 如果启动fail2ban报错](http://www.884358.com/webbench-test/#ru_guo_qi_dongfail2ban_bao_cuo)

[9 不起作用原因](http://www.884358.com/webbench-test/#bu_qi_zuo_yong_yuan_yin)

[9.1 测试匹配规则](http://www.884358.com/webbench-test/#ce_shi_pi_pei_gui_ze)

[9.2 检查fail2ban-python](http://www.884358.com/webbench-test/#jian_chafail2ban-python)

[10 使用CDN后获取真实IP地址](http://www.884358.com/webbench-test/#shi_yongCDN_hou_huo_qu_zhen_shiIP_de_zhi)

**文章目录 收缩**

* [WebBench安装](http://www.884358.com/webbench-test/#title-0)
* [WebBench使用](http://www.884358.com/webbench-test/#title-1)
* [控制并发数及请求频率](http://www.884358.com/webbench-test/#title-2)
* [安装fail2ban](http://www.884358.com/webbench-test/#title-3)
* [配置规则](http://www.884358.com/webbench-test/#title-4)
* [监控nginx访问日志](http://www.884358.com/webbench-test/#title-5)
* [常用命令](http://www.884358.com/webbench-test/#title-6)
* [如果启动fail2ban报错](http://www.884358.com/webbench-test/#title-7)
* [不起作用原因](http://www.884358.com/webbench-test/#title-8)
* [使用CDN后获取真实IP地址](http://www.884358.com/webbench-test/#title-9)

Webench是一款轻量级的网站测压工具，最多可以对网站模拟3w左右的并发请求，可以控制时间、是否使用缓存、是否等待服务器回复等等，且对中小型网站有明显的效果，基本上可以测出中小型网站的承受能力。

WebBench官网：http://home.tiscali.cz/~cz210552/webbench.html

测试环境说明：需要有两台服务器（A和B），在A服务器上上安装WebBench工具，对B服务器的网站进行测试。

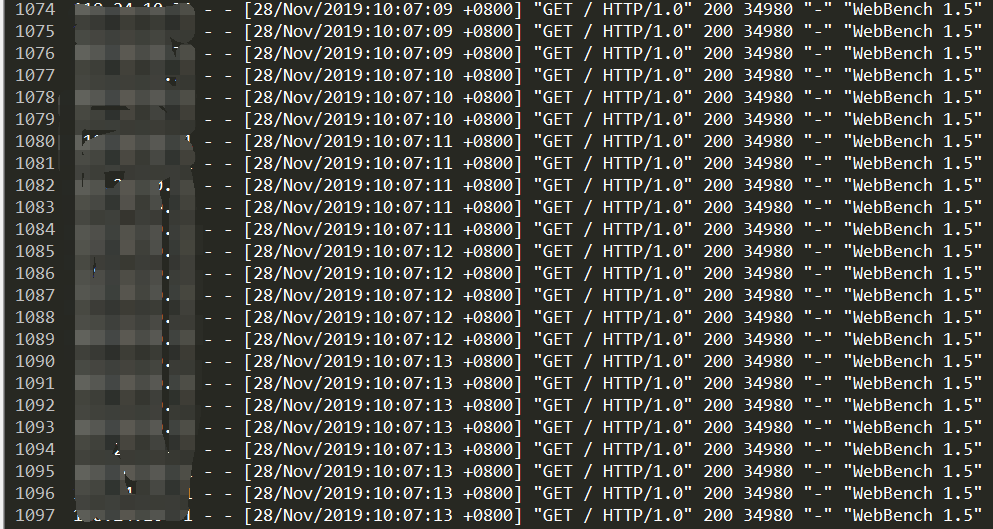
**WebBench安装**

1. yum install -y gcc ctags
2. wget http://file.884358.com/linux/webbench-1.5.tar.gz
3. tar zxvf webbench-1.5.tar.gz
4. cd webbench-1.5
5. make && make install

**WebBench使用**

1. webbench -c 100 -t 60 http://test.domain.com/phpinfo.php

webbench -c 并发数 -t 运行测试时间(秒) URL

通过测试，在A服务器执行webbench命令后，B服务器的CPU飙升至90%，同时B服务器操作卡顿，B服务器上的网站也打不开了。通过观察nignx日志，发现大量的并发请求：  


如何防止短时间的大量并发导致服务器宕机。  
解决办法：

*控制并发数，控制请求频率，必要时fail2ban*

**控制并发数及请求频率**

修改nginx配置，做如下设置：

1. ## 用户的 IP 地址 $binary\_remote\_addr 作为 Key，每个 IP 地址最多有 50 个并发连接
2. ## 你想开 几千个连接 刷死我？ 超过 50 个连接，直接返回 503 错误给你，根本不处理你的请求了
3. limit\_conn\_zone $binary\_remote\_addr zone=TotalConnLimitZone:10m ;
4. limit\_conn TotalConnLimitZone 50;
5. limit\_conn\_log\_level notice;
6. ## 用户的 IP 地址 $binary\_remote\_addr 作为 Key，每个 IP 地址每秒处理 10 个请求
7. ## 你想用程序每秒几百次的刷我，没戏，再快了就不处理了，直接返回 503 错误给你
8. limit\_req\_zone $binary\_remote\_addr zone=ConnLimitZone:10m rate=10r/s;
9. limit\_req\_log\_level notice;
10. ## 具体服务器配置
11. server {
12. listen 80;
13. location ~ \.php$ {
14. ## 最多 5 个排队， 由于每秒处理 10 个请求 + 5个排队，你一秒最多发送 15 个请求过来，再多就直接返回 503 错误给你了
15. limit\_req zone=ConnLimitZone burst=5 nodelay;
16. fastcgi\_pass 127.0.0.1:9000;
17. fastcgi\_index index.php;
18. include fastcgi\_params;
19. }
20. }

重启nginx，重新进行测试，测试结果发现A服务器的请求响应都变成了503：  
  
说明在nginx层面，已经拒绝了大量的并发请求。但尽管如此，服务器的CPU依然占用很高，网站依然打开缓慢。

**安装fail2ban**

**检查Firewalld是否启用**

1. firewall-cmd --state

fail2ban可以监控系统日志，并且根据一定规则匹配异常IP后使用Firewalld将其屏蔽，尤其是针对一些爆破/扫描等非常有效。

1. yum -y install fail2ban

安装成功后fail2ban配置文件位于/etc/fail2ban，其中jail.conf为主配置文件，相关的匹配规则位于filter.d目录，其它目录/文件一般很少用到，如果需要详细了解可自行搜索。

**配置规则**

新建jail.local来覆盖fail2ban的一些默认规则：

1. #新建配置
2. vi /etc/fail2ban/jail.local
3. #默认配置
4. [DEFAULT]
5. ignoreip = 127.0.0.1/8
6. bantime = 86400
7. findtime = 600
8. maxretry = 5
9. #这里banaction必须用firewallcmd-ipset,这是fiewalll支持的关键，如果是CentOS6,只有iptables那么填写：iptables-multiport
10. banaction = firewallcmd-ipset
11. action = %(action\_mwl)s

* ignoreip：IP白名单，白名单中的IP不会屏蔽，可填写多个以（,）分隔
* bantime：屏蔽时间，单位为秒（s）
* findtime：时间范围
* maxretry：最大次数
* banaction：屏蔽IP所使用的方法，上面使用firewalld屏蔽端口

**监控nginx访问日志**

这里仅以Nginx为例，使用fail2ban来监视nginx日志，匹配短时间内频繁请求的IP，并使用firewalld将其IP屏蔽，达到CC防护的作用。

1. #需要先新建一个nginx日志匹配规则
2. vi /etc/fail2ban/filter.d/nginx-cc.conf
3. #填写如下内容
4. [Definition]
5. failregex = <HOST> -.\*- .\*HTTP/[12].\* .\* .\*$
6. ignoreregex =

继续修改vi /etc/fail2ban/jail.local追加如下内容：

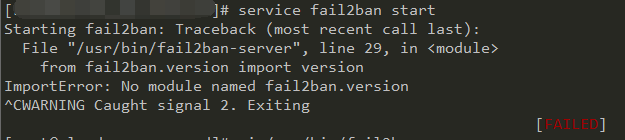
1. [nginx-cc]
2. enabled = true
3. port = http,https
4. filter = nginx-cc
5. action = %(action\_mwl)s
6. maxretry = 20
7. findtime = 60
8. bantime = 3600
9. logpath = /usr/local/nginx/logs/access.log

上面的配置意思是如果在60s内，同一IP达到20次请求，则将其IP ban 1小时，上面只是为了测试，请根据自己的实际情况修改。logpath为nginx日志路径。

**常用命令**

1. #启动
2. systemctl start fail2ban
3. #停止
4. systemctl stop fail2ban
5. #开机启动
6. systemctl enable fail2ban
7. #查看被ban IP，其中nginx-cc为名称
8. fail2ban-client status nginx-cc
9. #删除被ban IP
10. fail2ban-client set nginx-cc unbanip 192.168.111.111
11. #添加要ban的IP
12. fail2ban-client set nginx-cc banip 192.168.111.111
13. #查看日志
14. tail /var/log/fail2ban.log
15. #查看在fail2ban黑名单里的IP是否已经加到了firewall中。
16. ipset list

**如果启动fail2ban报错**

如果是在centos6.5上，启动fail2ban时有可能会报错，例如：  
  
有可能你升级过python的版本导致的，需要修改/usr/bin/fail2ban-client和/usr/bin/fail2ban-server这两个文件的第一行  
http://www.884358.com/wp-content/uploads/2020/03/049e52cefd99bd62c25b9c8cf9fe78b9.png

**不起作用原因**

正常情况，按照默认设置，如果我们一分钟内访问20次网页，就会被拉入黑名单，不允许访问了，如果经测试发现不起作用，可以通过如下办法来测试

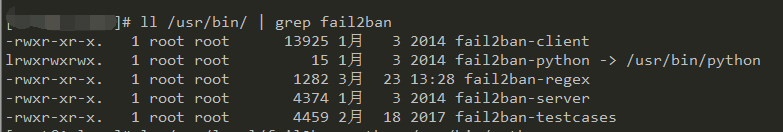
**测试匹配规则**

通过如下命令可以测试日志的匹配规则是否被命中

1. fail2ban-regex /usr/local/nginx/logs/access.log /etc/fail2ban/filter.d/nginx-cc.conf

**检查fail2ban-python**

1. ll /usr/bin/ | grep fail2ban

  
通过以上截图可以看到fail2ban-python是软链接到/usr/bin/python程序的，如果你是centos6.5且自己升级过python，那么需要重新设置下软链接

重新设置软链接

1. rm -rf /usr/bin/fail2ban-python
2. ln -s /usr/bin/python2.6 /usr/bin/fail2ban-python

**使用CDN后获取真实IP地址**

使用CDN后，通过日志可以看到IP都不是真实的，二是CDN的IP地址，如果继续屏蔽IP，只会把CDN的IP给屏蔽掉，而我们希望获取的是真实的IP地址。  
修改nginx配置文件，在http模块中添加如下代码：

1. ## 获取用户真实IP
2. log\_format main '$http\_x\_forwarded\_for - $remote\_user [$time\_local] "$request" '
3. '$status $body\_bytes\_sent "$http\_referer" '
4. '"$http\_user\_agent" ';

然后在每个站点的配置文件中，在access\_log后面加上main参数。例如宝塔的nginx站点配置文件是在：

1. /www/server/panel/vhost/nginx/xxx.conf

那么需要打开站点配置文件，在里面的access\_log后面加上main参数。例如：

1. access\_log /www/wwwlogs/xxx.log main;

同时，还需要修改之前的控制并发数及请求频率参数：

1. ## 这里取得原始用户的IP地址
2. map $http\_x\_forwarded\_for $clientRealIp {
3. "" $remote\_addr;
4. ~^(?P<firstAddr>[0-9\.]+),?.\*$ $firstAddr;
5. }
6. ## 针对原始用户 IP 地址做限制
7. limit\_conn\_zone $clientRealIp zone=TotalConnLimitZone:10m ;
8. limit\_conn TotalConnLimitZone 50;
9. limit\_conn\_log\_level notice;
10. ## 针对原始用户 IP 地址做限制
11. limit\_req\_zone $clientRealIp zone=ConnLimitZone:10m rate=10r/s;
12. #limit\_req zone=ConnLimitZone burst=10 nodelay; #如果开启此条规则，burst=10的限制将会在nginx全局生效
13. limit\_req\_log\_level notice;

重启nginx，然后再次观察日志文件，可以看到记录的IP是真实的IP了。

参考文章：  
Nginx在CDN加速之后，获取用户真实IP做并发访问限制的方法：https://zhang.ge/4879.html  
CentOS 7安装fail2ban + Firewalld防止爆破与CC攻击：https://www.xiaoz.me/archives/9831