Nginx

1.调度算法: 轮询(默认) weight(权重) ip\_hash(根据ip分配固定后端)

2.rewrit地址重写: 网页跳转 域名跳转 游览器跳转

1. upstream集群:

server ip:端口 weight=数 max\_fails=数 fail\_timeout=数 [down]

1. 优化(8个): 自定义报错页面、状态页面、增加并发量、数据包头缓存、

日志切割、页面压缩、服务器内存缓存(静态页面)

1. ulimit命令

临时:

ulimit -a 数 //查看所有属性值

ulimit -Hn 数 //设置硬限制（临时规则）

ulimit -Sn 数 //设置软限制（临时规则）

永久:

]# vim /etc/security/limits.conf

\* soft nofile 100000

用户或组 硬限制或软限制 需要限制的项目 限制的值

模块:

--with-http\_ssl\_module //开启SSL加密功能

--with-stream //开启TCP/UDP代理模块

--with-http\_stub\_status\_module //开启状态页面模块

?????cgi

########################################

状态页面

模块: 3个

location /status {

stub\_status on;

]# curl <http://192.168.4.5/status>

Active connections: 1

server accepts handled requests

10 10 3

Reading: 0 Writing: 1 Waiting: 0

----------------------------------------------------------------------------------

Active connections：当前活动的连接数量。

Accepts：已经接受客户端的连接总数量。

Handled：已经处理客户端的连接总数量。

（一般与accepts一致，除非服务器限制了连接数量）。

Requests：客户端发送的请求数量。

Reading：当前服务器正在读取客户端请求头的数量。

Writing：当前服务器正在写响应信息的数量。

Waiting：当前多少客户端在等待服务器的响应。

#########################################

用户认证

#######################################

rewrit地址重写

rewrite 旧地址 新地址 [选项]

last 不再读其他rewrite

break 不再读其他语句，完成本次请求后结束请求

redirect 临时重定向

permament 永久重定向

location / {

网页跳转 rewrite “/a.html$” /b.html;

域名跳转 rewrite ^/(.\*) <http://bbs.tarena.com/$1>;

游览器跳 if($http\_user\_agent~firefox){

rewrite ^(.\*)$ /nginx-firefox/$1 break;

}

upstream 集群池

http {

upstream 集群名 {

#ip\_hash;

server ip:端口 weight=数 max\_fails=数 fail\_timeout=数 [down];

#weight设置服务器权重值，默认值为1

#max\_fails设置最大失败次数

#fail\_timeout设置失败超时时间，单位为秒

#down标记服务器已关机，不参与集群调度

}

location / {

proxy\_connect\_timeout 1s; //连接的超时时间，可选配置

proxy\_timeout 3s;

proxy\_pass http://集群名;

}

}

#########################################

配置文件

http {

client\_header\_buffer\_size 1k; //默认请求包头信息的缓存

large\_client\_header\_buffers 4 4k; //大请求包头部信息的缓存个数与容量

location ~\* \.(jpg|jpeg|gif|png|css|js|ico|xml)$ {

expires 30d; //定义客户端缓存时间为30天

#########################################

Linux上Nginx理论上最大连接数

正向代理：最大连接数 = work\_processes \* worker\_connections

反向代理：最大连接数 = work\_processes \* worker\_connections / 4

备注：worker\_connections指单个进程能够处理的最大连接数，假设你的服务器是8核，worker\_connections = 65535（可以自定义的，但最大不能超过最大文件描述符数，因为1个连接相当于使用了1个文件描述符）

在上述2种情况下最大连接数（理论值）分别是： 8 \* 65535 和 8 \* 65535 /4

Linux上Haproxy理论上最大连接数

最大连接数 = maxconn( global | defaults | listen )