

nginx 反向代理和负载均衡

作者: [cmg](#)

原文链接: <https://ld246.com/article/1534945985472>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

负载均衡的目的是解决单个服务器压力过大，造成web响应过慢甚至服务器瘫痪，因为一个web服务同时处理用户的并发量有限。所以在几个服务器上部署同样的程序，实现集群。

模拟服务器集群可以在一台服务器上安装几个tomcat，修改每个tomcat的监听端口。（Server: port、Server > Service > Connector: port（HTTP/1.1协议）、Server > Service > Connector: port（AJP/1.3协议））

nginx负载均衡通过upstream模块实现，内置了3种策略：

- 轮询（默认）：根据请求次数，将每个请求均匀按权重分配到每台服务器
- 最少连接：将请求分配给当前连接数最少的服务器，Nginx会统计哪些服务器的连接数最少
- IP Hash：根据IP绑定处理请求的服务器。第一次请求时，根据该请求IP计算出Hash值，将请求分到集群中的某一台服务器上。以后来自改IP的所有请求，都将通过Hash算法，将请求转发到绑定的服务器上

upstream模块

server指令配置处理请求的服务器IP或域名,端口可选，可以追加其他配置参数：

weight: 权重，数字越大,权重越大,被分配请求的机会越大，默认为1

max_fails 允许请求失败的次数,超过该次数后,在fail_imeout的时间之内,新的请求将不会分配到该机器,默认为1

fail_timeout: max_fails次失败后，暂停服务时间，默认为10s

backup: 备份机,所有其他服务器都不可用后才会生效

down: 将指定服务器标示为不可用

max_conns: 服务器最大连接数,超过后将不会分配新的请求,默认为0,表示不限制

```
http {
    proxy_connect_timeout 5;
    proxy_read_timeout 60;
    proxy_send_timeout 5;
    proxy_buffer_size 16k;
    proxy_buffers 4 64k;
    proxy_busy_buffers_size 128k;
    proxy_temp_file_write_size 128k;
```

设置缓存临时目录

```
proxy_temp_path /data/proxy_temp_dir;
```

设置缓存目录，并设置Web缓存区名称为cache_one，内存缓存空间大小为200MB，1天没有被访的内容自动清除，硬盘缓存空间大小为30GB

```
proxy_cache_path /data/proxy_cache_dir levels=1:2 keys_zone=cache_one:200m inactive=1d
max_size=30g;
```

设置代理的后端tomcat集群

```
upstream web_server {
    server 192.168.0.102:8080 weight=1 max_fails=2 fail_timeout=30s;
    server 192.168.0.103:8080 weight=1 max_fails=2 fail_timeout=30s;
    server 192.168.0.104:8080 weight=1 max_fails=2 fail_timeout=30s;
}
```

```

server {

    location {

        proxy_pass http://web_server; # 代理的集群名称
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
        # 如果后端的服务器返回502、504、执行超时等错误，自动将请求转发到upstream负载均衡池
        # 的另一台服务器，实现故障转移
        proxy_next_upstream http_502 http_504 error timeout invalid_header;
        proxy_cache cache_one;
        # 对不同的HTTP状态码设置不同的缓存时间
        proxy_cache_valid 200 304 12h;
        # 以域名、URI、参数组合成Web缓存的Key值，Nginx根据Key值哈希，存储缓存内容到二级缓存
        # 目录内
        proxy_cache_key $host$uri$is_args$args;
    }

    # 清除缓存，访问http://192.168.0.101/purge/shop/hadoop.html来清除http://192.168.0.101/s
    # op/hadoop.html的页面缓存
    location ~ /purge/(.*) {
        allow 127.0.0.1;
        allow 192.168.0.0/16;
        deny all;
        proxy_cache_purge cache_one $host$1$is_args$args;
    }

    # 扩展名以.php、.jsp、.cgi、.jhtml结尾的动态应用程序不缓存
    location ~ .*\.?(php|jsp|cgi|jhtml)?$ {
        proxy_pass http://web_server;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_cache cache_one;
        proxy_cache_valid 200 304 12h;
        proxy_cache_key $host$uri$is_args$args;
    }

}

```