
[博客园](#)
[首页](#)
[新随笔](#)
[联系](#)
[订阅](#)
[管理](#)

## Nginx代理的几种模式

### Nginx代理的几种模式

#### 目录

- 一、正向代理(Forward Proxy)
- 二、反向代理 (reverse proxy)
- 三、透明代理

转载自一位大佬

通常我们都知道Nginx性能很高，尤其是作为一个代理服务器，因为它用的是epoll模型，就比如Python Django Web的性能不行，我们可能就会在前端加一个nginx代理，从而提高总体的处理性能问题，代理服务技术是在互联网早期就出现被使用的。一般实现代理技术的方式就是在服务器上安装代理服务软件，让其成为一个代理服务器，从而实现代理技术。常用的代理技术分为正向代理、反向代理和透明代理。

#### 一、正向代理(Forward Proxy)

一般情况下，如果没有特别说明，代理技术默认说是正向代理技术。

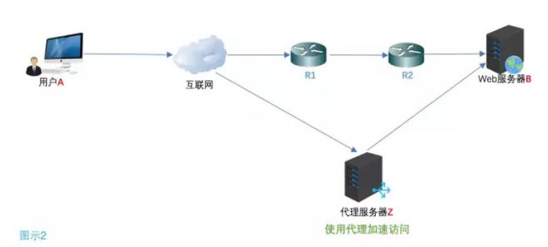
正向代理 (forward) 是一个位于客户端【用户A】和原始服务器(origin server)【服务器B】之间的服务器【代理服务器Z】，为了从原始服务器取得内容，用户A向代理服务器Z发送一个请求并指定目标(服务器B)，然后代理服务器Z向服务器B转交请求并将获得的内容返回给客户端。客户端必须要进行一些特别的设置才能使用正向代理，如下图所示：



从上图看，所谓的正向代理就是代理服务器【Z】替代访问方【用户A】去访问目标服务器【服务器B】。

下面就是正向代理的意义以及场景

#### 1、用户访问本来无法访问的服务器B的资源



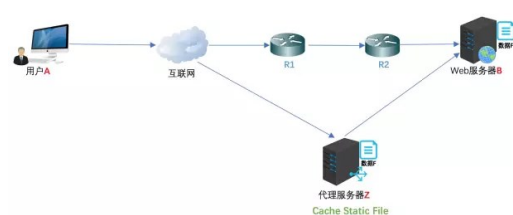
假设最初用户A要访问服务器B需要经过R1和R2路由器这样一个路由节点，如果路由器R1或者路由器R2发生故障，那么就无法访问服务器B了。但是如果用户A让代理服务器Z去代替自己访问服务器B，由于代理服务器Z没有经过路由器R1或R2节点中，而是通过其它的路由节点访问服务器B，那么用户A就可以得到服务器B的数据了。

现实中的例子就是“FQ”，不过自从VPN技术被广泛应用外，“FQ”不但使用了传统的正向代理技术，有的还使用了VPN技术。例如IP-SECT动态vpn技术等

#### 2.加速访问服务器B资源

这种说法目前不像以前那么流行了，主要是带宽流量的飞速发展。早期的正向代理中，很多人使用正向代理就是提速。还是如图2假设用户A到服务器B，经过R1路由器和R2路由器，而R1到R2路由器的链路是一个低带宽链路。而用户A到代理服务器Z，从代理服务器Z到服务器B都是高带宽链路。那么很显然就可以加速访问服务器B了。

#### 3.Cache作用



Cache（缓存）技术和代理服务技术是紧密联系的（不光是正向代理，反向代理也使用了Cache（缓存）技术。如上图所示，如果在用户A访问服务器B某数据F之前，已经有人通过代理服务器Z访问过服务器B上得数据F，那么代理服务器Z会把数据F保存一段时间，如果有人正好取该数据F，那么代理服务器Z不再访问服务器B，而把缓存的数据F直接发给用户A。这一技术在Cache中术语就叫Cache命中。如果有更多的像用户A的用户来访问代理服务器Z，那么这些用户都可以直接从代理服务器Z中取得数据F，而不用千里迢迢的去服务器B下载数据了。

#### 4.客户端访问授权

这方面的内容现今使用的还是比较多的，例如一些公司采用ISA Server做为正向代理服务器来授权用户是否有权访问互联网，如下图所示：

### 公告

昵称： 阿东123  
 园龄： 4年5个月  
 粉丝： 3  
 关注： 16  
 +加关注

< 2022年11月 >						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

### 搜索





### 最新随笔

- 1.linux上执行dll
- 2.Python 打包exe
- 3.Git 版本控制 添加文件/提交
- 4.Jeecg-boot开发环境搭建
- 5.IntelliJ IDEA 2021破解
- 6.yarn : 无法加载文件 C:\Program Files\nodejs\node\_global\yarn.ps1，因为在此系统上禁止运行脚本。
- 7.SQL Server如何启用xp\_cmdshell组件
- 8.Net6 应用程序运行在Linux系统
- 9.码云Gitee客户端配置
- 10.zabbix中文显示乱码解决

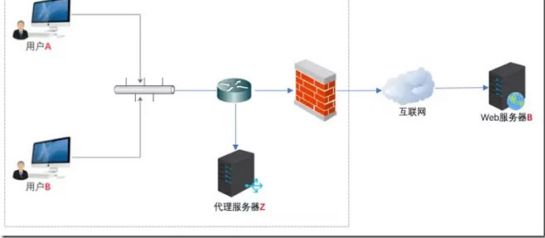
### 我的标签

- c#(16)
- ASP.NET(15)
- SQL SERVER(8)
- JS(7)
- WinCC(7)
- vue(6)
- jQuery(5)
- zabbix(4)
- LINUX(3)
- HTML(2)
- 更多

### 积分与排名

积分 - 36739  
 排名 - 39918

### 随笔档案



如上图防火墙作为网关，用来过滤外网对其的访问。假设用户A和用户B都设置了代理服务器，用户A允许访问互联网，而用户B不允许访问互联网（这个在代理服务器Z上做限制）这样用户A因为授权，可以通过代理服务器访问到服务器B，而用户B因为没有被代理服务器Z授权，所以访问服务器B时，数据包会被直接丢弃。

5.隐藏访问者的行踪

如下图所示，我们可以看出服务器B并不知道访问自己的实际是用户A，因为代理服务器Z代替用户A去直接与服务器B进行交互。如果代理服务器Z被用户A完全控制（或不完全控制），会惯以“肉鸡”术语称呼。



总结：

正向代理是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端必须设置正向代理服务器，当然前提是要知道正向代理服务器的IP地址，还有代理程序的端口。

Nginx正向代理配置示例如下：

```
server{
    resolver 8.8.8.8;
    resolver_timeout 30s;
    listen 82;
    location / {
        proxy_pass http://$http_host$request_uri;
        proxy_set_header Host $http_host;
        proxy_buffers 256 4k;
        proxy_max_temp_file_size 0;
        proxy_connect_timeout 30;
        proxy_cache_valid 200 302 10m;
        proxy_cache_valid 301 1h;
        proxy_cache_valid any 1m;
    }
}
```

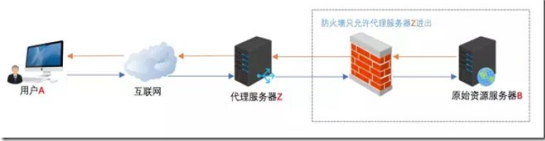
1、不能有hostname。  
2、必须有resolver，即dns，即上面的8.8.8.8，超时时间(30秒)可选。  
3、配置正向代理参数，均是由 Nginx 变量组成。  
proxy\_pass \$scheme://\$host\$request\_uri;  
proxy\_set\_header Host \$http\_host;  
4、配置缓存大小，关闭磁盘缓存读写减少I/O，以及代理连接超时时间。  
proxy\_buffers 256 4k;  
proxy\_max\_temp\_file\_size 0;  
proxy\_connect\_timeout 30;  
5、配置代理服务器 Http 状态缓存时间。  
proxy\_cache\_valid 200 302 10m;  
proxy\_cache\_valid 301 1h;  
proxy\_cache\_valid any 1m;  
配置好后，重启nginx。以浏览器为例，要使用这个代理服务器，则只需将浏览器代理设置为http://+服务器ip:地址+82(82是刚刚设置的端口号)即可使用了。

二、反向代理 (reverse proxy)

反向代理正好与正向代理相反，对于客户端而言代理服务器就像是原始服务器，并且客户端不需要进行任何特别的设置。客户端向反向代理的命名空间(name-space)中的内容发送普通请求，接着反向代理将判断向何处(原始服务器)转交请求，并将获得的内容返回给客户端。

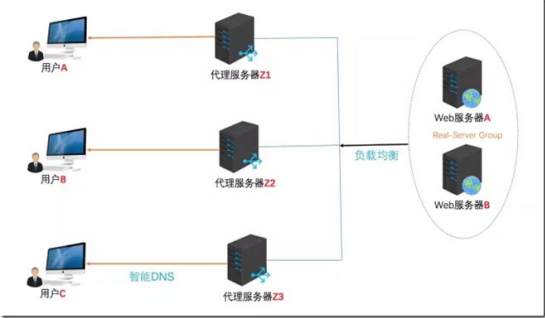
使用反向代理服务器的作用如下：

1.保护和隐藏原始资源服务器



用户A始终认为它访问的是原始服务器B而不是代理服务器Z，但实际上反向代理服务器接受用户A的应答，从原始资源服务器B中取得用户A的需求资源，然后发送给用户A。由于防火墙的作用，只允许代理服务器Z访问原始资源服务器B。尽管在这个虚拟的环境下，防火墙和反向代理的共同作用保护了原始资源服务器B，但用户A并不知道。

2.负载均衡

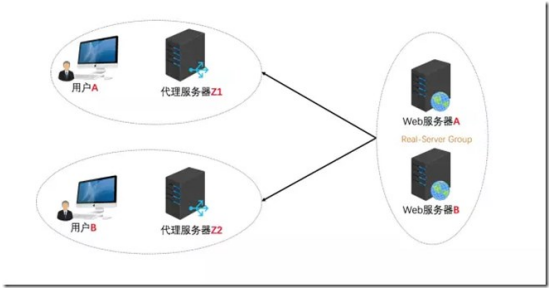


当反向代理服务器不止一个的时候，我们甚至可以把它们做成集群，当更多的用户访问资源服务器B的时候，让不同的代理服务器Z (x) 去应答不同的用户，然后发送不同用户需要的资源。  
当然反向代理服务器像正向代理服务器一样拥有Cache的作用，它可以缓存原始资源服务器B的资源，而不是每次都要向原始资源服务器组请求数据，特别是一些静态的数据，比如图片和文件，如果这些反向代理服务器能够做到和用户X来自同一个网络，那么用户X

2022年9月(3)
2022年8月(16)
2022年7月(7)
2022年5月(2)
2022年4月(4)
2022年2月(2)
2021年12月(3)
2021年5月(2)
2021年2月(2)
2020年12月(1)
2020年11月(17)
2020年9月(1)
2020年4月(4)
2020年3月(11)
2020年2月(5)
更多

阅读排行榜
1. select框内改变时触发事件的实现(12332)
2. c#之task与thread区别及其使用(8229)
3. 前端js几种加密/解密方法(8103)
4. Nginx代理的几种模式 (2384)
5. C#调用存储过程(1986)

访问反向代理服务器X，就会得到很高质量的速度。这正是CDN技术的核心。如下图所示：



反向代理理论与正向代理正好相反，对于客户端而言它就像是原始服务器，并且客户端不需要进行任何特别的设置。客户端向反向代理的命名空间(name-space)中的内容发送普通请求，接着反向代理将判断向何处(原始服务器)转交请求，并将获得的内容返回给客户端，就像这些内容原本就是它自己的一样。

基本上，网上做正向代理的程序很多，能做正向代理的软件大部分也可以做反向代理。开源软件中最流行的就是squid，既可以做正向代理，也有很多人用来做反向代理的前端服务器。另外MS ISA也可以用来在Windows平台下做正向代理。反向代理中最主要的实践就是WEB服务，近些年来最火的就是Nginx了。网上有人说Nginx不能做正向代理，其实是不对的。Nginx也可以做正向代理，不过用的人比较少了。

#### Nginx反向代理示例：

```
http {
# 省略了前面一般的配置, 直接从负载均衡这里开始
# 设置地址池, 后端:台服务器
upstream http_server_pool {
server 192.168.1.2:8080 weight=2 max_fails=2 fail_timeout=30s;
server 192.168.1.3:8080 weight=3 max_fails=2 fail_timeout=30s;
server 192.168.1.4:8080 weight=4 max_fails=2 fail_timeout=30s;
}
# 一个虚拟主机, 用来反向代理http_server_pool这组服务器
server {
listen 80;
# 外网访问的域名
server_name www.test.com;
location / {
# 后端服务器返回500 503 404错误, 自动请求转发到upstream池中另一台服务器
proxy_next_upstream error timeout invalid_header http_500 http_503 http_404;
proxy_pass http://http_server_pool;
proxy_set_header Host www.test.com;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
access_log logs/www.test.com.access.log combined;
}
}

最简单的反向代理演示(在一台服务器上做代理服务器, 将http请求转发到另一台iis服务器上, 通过二级域名形式访问。)编辑vim nginx.conf
server {
listen 80;
server_name test.zhoumengkang.com;
location / {
proxy_pass http://121.199.*.*:80;
}
}

参考:http://www.blogjava.net/xiaomage234/archive/2011/09/08/358247.html
```

### 三、透明代理

透明代理的意思是客户端根本不需要知道有代理服务器的存在，它改编你的request fields（报文），并会传递真实IP。注意，加密的透明代理则是属于匿名代理，意思是不用设置使用代理了。

透明代理实践的例子就是时下很多公司使用的行为管理软件。如下图所示：



用户A和用户B并不知道行为管理设备充当透明代理行为，当用户A或用户B向服务器A或服务器B提交请求的时候，透明代理设备根据自身策略拦截并修改用户A或B的报文，并作为实际的请求方，向服务器A或B发送请求。当接收信息回传，透明代理再根据自身的设置把允许的报文发回至用户A或B，如上图，如果透明代理设置不允许访问服务器B，那么用户A或者用户B就不会得到服务器B的数据。

#### Nginx透明代理配置示例：

```
# cat /etc/nginx/sites-enabled/proxy
server {
resolver 8.8.8.8;
access_log off;
listen [::]:8080;
location / {
proxy_pass $scheme://$host$request_uri;
proxy_set_header Host $http_host;
proxy_buffers 256 4k;
proxy_max_temp_file_size 0k;
}
}

iptables -t nat -A PREROUTING -s 10.8.0.0/24 -p tcp --dport 80 -j DNAT --to 192.168.0.253:8080
RAW Paste Data
# cat /etc/nginx/sites-enabled/proxy
server {
resolver 8.8.8.8;
access_log off;
listen [::]:8080;
location / {
proxy_pass $scheme://$host$request_uri;
proxy_set_header Host $http_host;
proxy_buffers 256 4k;
proxy_max_temp_file_size 0k;
}
}

iptables -t nat -A PREROUTING -s 10.8.0.0/24 -p tcp --dport 80 -j DNAT --to 192.168.0.253:8080
```



标签: Nginx

好文置顶

关注我

收藏此文

🔥

🗨️



阿东123  
粉丝 · 3 关注 · 16

+ 加关注

0

👍 推荐

0

👎 反对

« 上一篇: Nginx配置与命令

» 下一篇: QT6 发布程序

posted @ 2022-02-28 11:16 阿东123 阅读(2395) 评论(0) 编辑 收藏 举报

刷新评论 刷新页面 返回顶部



登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐:

- 定制 ASP.NET Core 的身份认证
- 从 WinDbg 角度理解 .NET7 的 AOT 玩法
- 当 xxl-job 遇上 docker → 它暴了，我也乱了！
- Go map 竟然也会发生内存泄漏？
- .net 温故知新：.NET日志记录 ILogger使用和原理

阅读排行:

- .NET周报【11月第2期 2022-11-15】
- 两万字盘点被玩烂了的9种设计模式
- 网络工程师必备神器 --- 千月网络助手（附下载链接）
- 二维码的秘密（生成原理）
- 同事吐槽我的接口性能差，原来它是真凶！