### 走一边

systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

# 什么是Nginx？为什么使用Nginx？

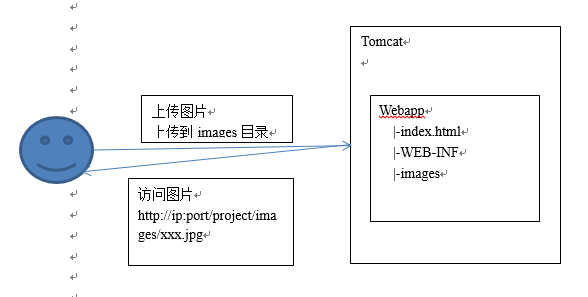
一、前言

      为毛要用nginx服务器代理，不直接用tomcat 7.0，还做多了一次接请求？

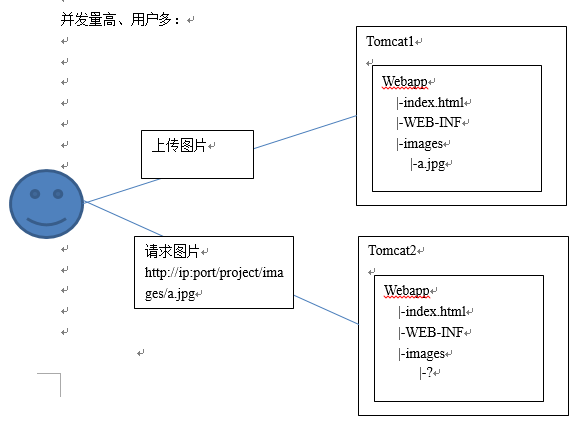
      这个是我想问的，公司的新项目是要用Nginx+tomcat7+jdk开发的，用户命名可以直接访问tomcat，为啥还要用Nginx？这货是个啥玩意？

二、为什么使用Nginx？

      在传统的Web项目中，并发量小，用户使用的少。所以在低并发的情况下，用户可以直接访问tomcat服务器，然后tomcat服务器返回消息给用户。比如，我们上传图片：



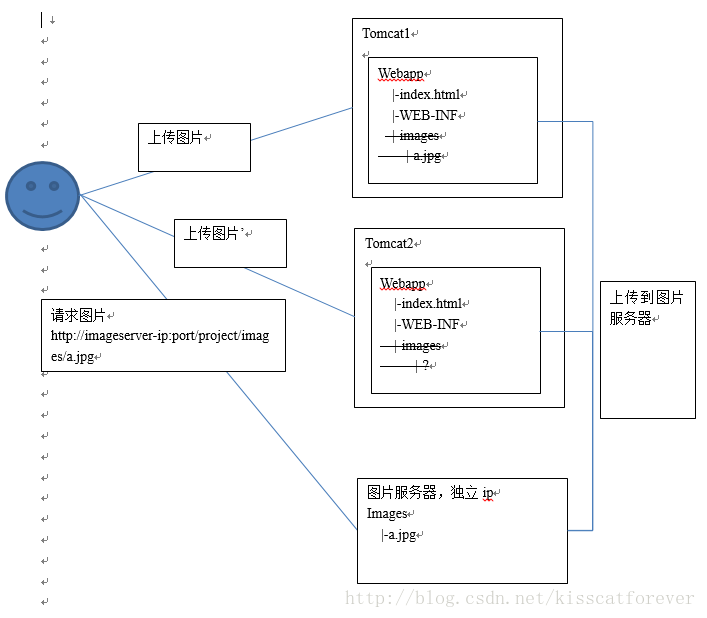
      当然我们知道，为了解决并发，可以使用负载均衡：也就是我们多增加几个tomcat服务器。当用户访问的时候，请求可以提交到空闲的tomcat服务器上。



      但是这种情况下可能会有一种这样的问题：上传图片操作。我们把图片上传到了tomcat1上了，当我们要访问这个图片的时候，tomcat1正好在工作，所以访问的请求就交给其他的tomcat操作，而tomcat之间的数据没有进行同步，所以就发生了我们要请求的图片找不到。

      为了解决这种情况，我们就想出了分布式。我们专门建立一个图片服务器，用来存储图片。这样当我们都把图片上传的时候，不管是哪个服务器接收到图片，都把图片上传到图片服务器。

      图片服务器上需要安装一个http服务器，可以使用tomcat、apache、nginx。



      看到这里大家可能会问，既然我们要选择的是http服务器，为什么不继续使用tomcat，而要使用Nginx？

      原因如下：nginx常用做静态内容服务和代理服务器（不是你FQ那个代理），直面外来请求转发给后面的应用服务（tomcat，django什么的），tomcat更多用来做做一个应用容器，让java web app跑在里面的东西，对应同级别的有jboss,jetty等东西。

三、什么是Nginx？

      根据前面的对比，我们可以了解到Nginx是一个http服务器。是一个使用c语言开发的高性能的http服务器及反向代理服务器。Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师Igor Sysoev所开发，官方测试nginx能够支支撑5万并发链接，并且cpu、内存等资源消耗却非常低，运行非常稳定。

3.1 Nginx的应用场景

      1、 http服务器。Nginx是一个http服务可以独立提供http服务。可以做网页静态服务器。

      2、 虚拟主机。可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。

基于端口的，不同的端口

基于域名的，不同域名

      3、 反向代理，负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用nginx做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

四、小结

      通过了解Nginx也算是多了一门技术，而且还是挺简单的，使用也比较方便。重点是可以给系统带来很大的性能提升。

      下一篇将讲解如何安装Nginx。

# [nginx和tomcat的区别](https://www.cnblogs.com/flypie/p/5153702.html)

web上的server都叫web server，但是大家分工也有不同的。

nginx常用做静态内容服务和代理服务器（不是你FQ那个代理），直面外来请求转发给后面的应用服务（tomcat，django什么的），tomcat更多用来做做一个应用容器，让java web app跑在里面的东西，对应同级别的有jboss,jetty等东西。  
  
但是事无绝对，nginx也可以通过模块开发来提供应用功能，tomcat也可以直接提供http服务，通常用在内网和不需要流控等小型服务的场景。  
  
apache用的越来越少了，大体上和nginx功能重合的更多。

严格的来说，Apache/Nginx 应该叫做「HTTP Server」；而 Tomcat 则是一个「Application Server」，或者更准确的来说，是一个「Servlet/JSP」应用的容器（Ruby/Python 等其他语言开发的应用也无法直接运行在 Tomcat 上）。

一个 HTTP Server 关心的是 HTTP 协议层面的传输和访问控制，所以在 Apache/Nginx 上你可以看到代理、负载均衡等功能。客户端通过 HTTP Server 访问服务器上存储的资源（HTML 文件、图片文件等等）。通过 CGI 技术，也可以将处理过的内容通过 HTTP Server 分发，但是一个 HTTP Server 始终只是把服务器上的文件如实的通过 HTTP 协议传输给客户端。  
  
而应用服务器，则是一个应用执行的容器。它首先需要支持开发语言的 Runtime（对于 Tomcat 来说，就是 Java），保证应用能够在应用服务器上正常运行。其次，需要支持应用相关的规范，例如类库、安全方面的特性。对于 Tomcat 来说，就是需要提供 JSP/Sevlet 运行需要的标准类库、Interface 等。为了方便，应用服务器往往也会集成 HTTP Server 的功能，但是不如专业的 HTTP Server 那么强大，所以应用服务器往往是运行在 HTTP Server 的背后，执行应用，将动态的内容转化为静态的内容之后，通过 HTTP Server 分发到客户端。

# 使用Nginx做负载均衡

一、前言

      前文简单介绍了如何安装Nginx，方法非常简单，操作也非常方便。上文中提到了Nginx可以做http服务器、虚拟主机、负载均衡。在这篇博客中小编就向大家介绍一下如何用Nginx做负载均衡。

二、负载均衡

      首先我们得知道什么是负载均衡。

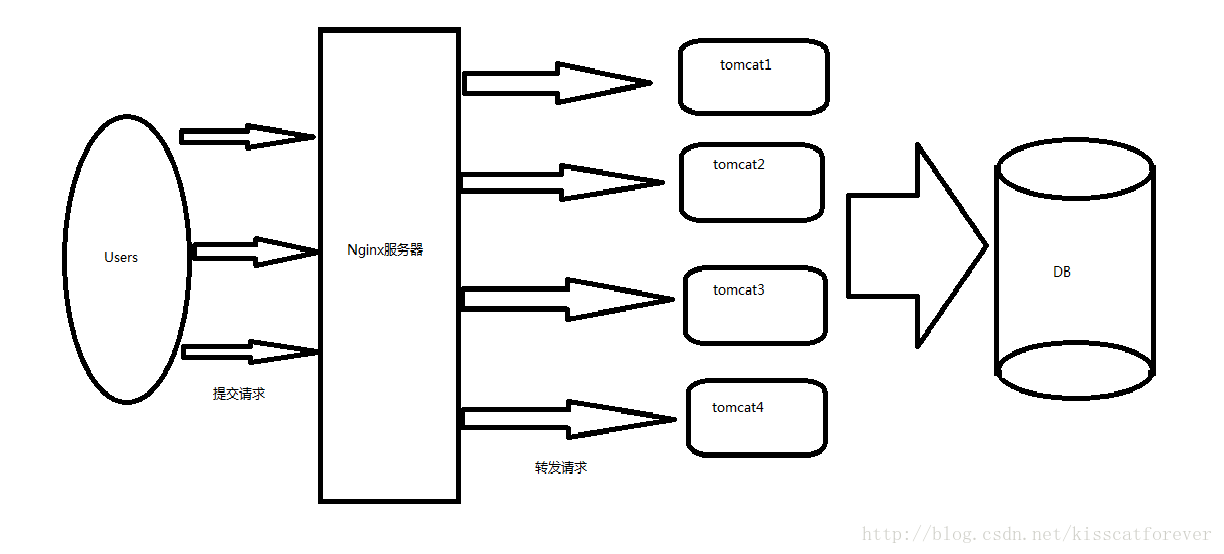
      负载均衡 建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

      负载均衡，英文名称为Load Balance，其意思就是分摊到多个操作单元上进行执行，例如Web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。【百度百科】

负载均衡的核心是：**将负载进行平衡**

三、Nginx 负载均衡

      对于Nginx来说，他可以把提交的请求进行转发。架构如下：



架构图解释：在这个图中，用户提交请求先到Nginx服务器，然后由Nginx进行对这些请求的转发，转发到各个tomcat服务器上，然后tomcat进行逻辑处理。

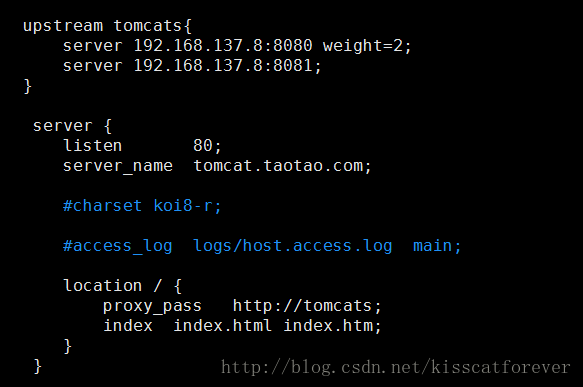
      而Nginx是如何做的呢？首先Nginx中写入了各个tomcat的服务器的地址，架构师可以根据tomcat的性能，增加对其使用的程度。Nginx只需在nginx.conf中进行配置：

      只需要在upostream的server后面添加一个weight，即可代表权重。权重越大，分配请求的数量就越多。不写的时候默认是1。

      配置流程如下：

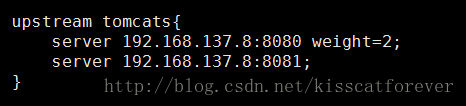
1.Nginx的配置

      在nginx中配置一个server：使用域名的方式来进行反向代理，这里使用的域名是tomcat.taotao.com,用户可以访问这个域名，先提交到Nginx中。



2.tomcat配置

      开启两台tomcat，端口如下：



Nginx根据proxy\_pass来访问的路径来确定访问的server，根据权重自动分配选择的哪一个。权重大的处理的概率高。

四、小结

      通过本次的学习Nginx的负载均衡，重点是引出了一个权重，权重越大就处理的请求越多。

      好了，下一篇博客将向大家介绍Nginx的反向代理的功能。

# 我们选择Nginx的理由?

1:更快

2:高扩展性

3:高可靠性

4:低内存消耗

5:单机支持10W以上的并发连接

6:热部署

7:最自由的BSD许可协议

# 使用Nginx必备的软件

1:GCC编译器, G++编译器

yum install -y gcc

yum install -y gcc-c++

2:PCRE库(Perl兼容正则表达式)

yum install -y pcre pcre-devel

3:zlib库

yum install -y zlib zlib-devel

4:OpenSSL开发库

yum install -y openssl openssl-devel

磁盘目录:

部署目录,默认情况下,该目录为/usr/local/nginx

编译安装Nginx

./configure

make

make install

# Nginx的命令行控制

1:默认方式启动

/usr/local/nginx/sbin/nginx

2:另行指定配置文件的启动方式

/usr/local/nginx/sbin/nginx -c /tmp/nginx.conf

3:另行指定安装目录的启动方式

/usr/local/nginx/sbin/nginx -p /usr/local/nginx/

4:另行指定全局配置项的启动方式

/usr/local/nginx/sbin/nginx -g “pid /var/nginx/test.pid;”

如果要停止Nginx服务,那么需要执行下面代码

/usr/local/nginx/sbin/nginx -g “pid /var/nginx/test.pid;” -s stop

5:测试配置信息是否有错误

在不启动Nginx的情况下,使用-t参数仅测试配置文件是否有错误

/usr/local/nginx/sbin/nginx -t

6:在测试配置阶段不输出信息

/usr/local/nginx/sbin/nginx -t -q

7:显示版本信息

/usr/local/nginx/sbin/nginx -v

8:显示编译阶段的参数

/usr/local/nginx/sbin/nginx -V

9:快速地停止服务

/usr/local/nginx/sbin/nginx -s stop

10:优雅地停止服务

/usr/local/nginx/sbin/nginx -s quit

11:使运行中的Nginx重读配置项并生效

/usr/local/nginx/sbin/nginx -s reload

12:日志回滚

/usr/local/nginx/sbin/nginx -s reopen

# 用HTTP proxy module配置一个反向代理服务器

反向代理（reverse proxy）方式是指用代理服务器来接受Internet上的连接请求，然后将

请求转发给内部网络中的上游服务器，并将从上游服务器上得到的结果返回给Internet上请求

连接的客户端，此时代理服务器对外的表现就是一个Web服务器。充当反向代理服务器也是

Nginx的一种常见用法（反向代理服务器必须能够处理大量并发请求），本节将介绍Nginx作

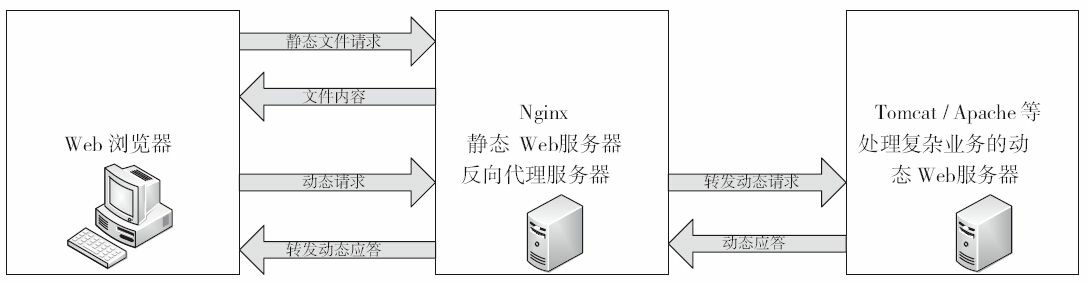
为HTTP反向代理服务器的基本用法。

由于Nginx具有“强悍”的高并发高负载能力，因此一般会作为前端的服务器直接向客户

端提供静态文件服务。但也有一些复杂、多变的业务不适合放到Nginx服务器上，这时会用

Apache、Tomcat等服务器来处理。于是，Nginx通常会被配置为既是静态Web服务器也是反向

代理服务器（如图2-3所示），不适合Nginx处理的请求就会直接转发到上游服务器中处理。

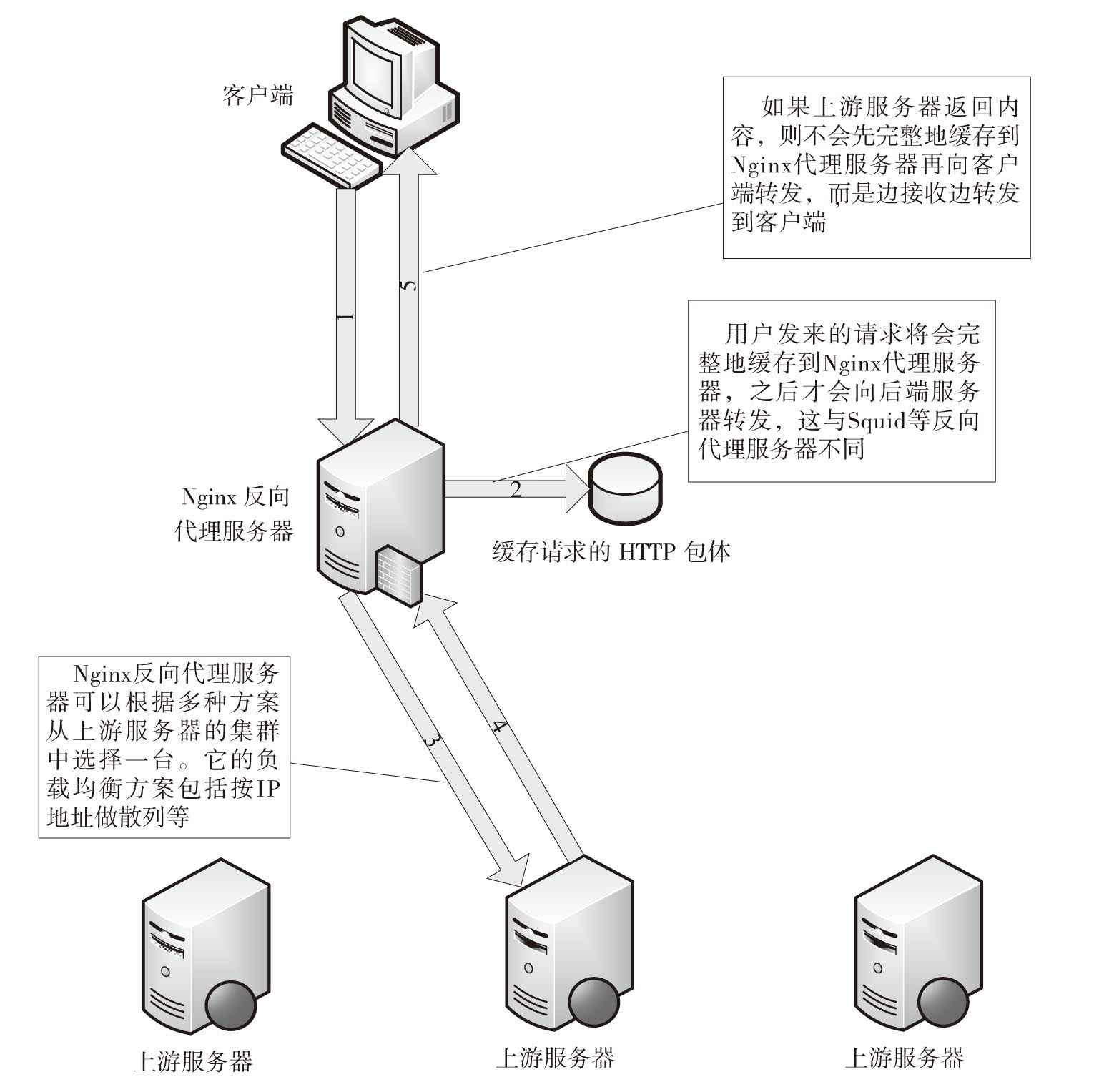


作为静态Web服务器与反向代理服务器的Nginx

Nginx的这种工作方式有什么优缺点呢？很明显，缺点是延长了一个请求的处理时间，

并增加了用于缓存请求内容的内存和磁盘空间。而优点则是降低了上游服务器的负载，尽量

把压力放在Nginx服务器上。



Nginx作为反向代理服务器时转发请求的流程