http://www.food.com/xxxx/xxx

We can break the url into three parts,

The part before colon slash slash is what we call the URL scheme

The last part is url path



P.S.: The **Domain Name System** (**DNS**) is a [hierarchical](https://en.wikipedia.org/wiki/Hierarchical) decentralized naming system for computers, services, or other resources connected to the [Internet](https://en.wikipedia.org/wiki/Internet) or a private network. It associates various information with [domain names](https://en.wikipedia.org/wiki/Domain_name) assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical [IP addresses](https://en.wikipedia.org/wiki/IP_address) needed for locating and identifying computer services and devices with the underlying network protocols.

I got some texts describing Cache-control:

# [浅谈前端性能优化（一）——Expires和Cache-Control](https://blog.csdn.net/zhouziyu2011/article/details/71312452)

标签： [前端性能优化](http://so.csdn.net/so/search/s.do?q=前端性能优化&t=blog) [Expires](http://so.csdn.net/so/search/s.do?q=Expires&t=blog) [Cache-Control](http://so.csdn.net/so/search/s.do?q=Cache-Control&t=blog)

2017年05月07日 11:57:05 3724人阅读 [评论](javascript:void(0);)(0) [收藏](javascript:void(0);) [举报](javascript:void(0);)

https://csdnimg.cn/release/phoenix/images/category_icon.jpg分类：

Web（47） https://csdnimg.cn/release/phoenix/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/zhouziyu2011/article/details/71312452

前端性能优化中，减少HTTP请求可以提高页面的响应速度。

浏览器在第一次访问页面时向服务器请求资源，并缓存起来，下次再访问时会判断在缓存中是否已有该资源且有没有更新过，如果已有该资源且未更新过，则直接从浏览器缓存中读取。原理：通过HTTP 请求头中的 If-Modified-Since（If-No-Match） 和响应头中的Last-Modified（ETag）来实现，HTTP请求把 If-Modified-Since（If-No-Match）传给服务器，服务器将其与Last-Modified（ETag）对比，若相同，则文件没有被改动过，则返回304，直接浏览器缓存中读取资源即可。

问题：虽然该方法减少了已缓存资源的下载时间，但仍然发起了一次http请求。

解决：已缓存资源不再发起http请求，即HTTP的Expires和Cache-Control。对一个网站而言，CSS、JavaScript、图片等静态资源更新的频率都比较低，而这些文件又几乎是每次HTTP请求都需要的，如果将这些文件缓存在浏览器中，可以极好的改善性能。通过设置http头中的cache-control和expires的属性，可设定浏览器缓存，将静态内容设为永不过期，或者很长时间后才过期。

1、Cache-Control

Cache-Control属性是在服务器端配置的，不同的服务器有不同的配置，apache、nginx、IIS、tomcat等配置都不尽相同。

以Apache为例，在http.conf中做如下配置：

**[plain]** [view plain](https://blog.csdn.net/zhouziyu2011/article/details/71312452) [copy](https://blog.csdn.net/zhouziyu2011/article/details/71312452)

1. <filesMatch ”.(jpg|jpeg|png|gif|ico)$”>
2. Header set Cache Control max-age=16768000,public
3. </filesMatch>
4. <filesMatch ”.(css|js)$”>
5. Header set Cache Control max-age=2628000,public
6. </filesMatch>

问题：浏览器缓存的资源，若又想更新资源，如何实现？

解决：通过修改该资源的名称来实现。修改了资源名称，浏览器会当做不同的资源。

2、Expires

Expires属性也是在服务端配置的，具体的配置也根据服务器而定。

问题：可能存在客户端时间跟服务端时间不一致的问题。

解决：建议Expires结合Cache-Control一起使用。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*代理

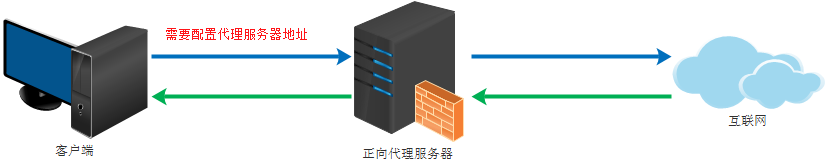
# [正向代理与反向代理【总结】](http://www.cnblogs.com/Anker/p/6056540.html)

**1、前言**

　　最近工作中用到反向代理，发现网络代理的玩法还真不少，网络背后有很多需要去学习。而在此之前仅仅使用了过代理软件，曾经为了访问google，使用了代理软件，需要在浏览器中配置代理的地址。我只知道有代理这个概念，并不清楚代理还有正向和反向之分，于是赶紧学习一下，补充一下知识。首先弄清楚什么是正向代理，什么是反向代理，然后是二者在实际使用中展示的方式是什么样的，最后总结一下正向代理用来做什么，反向代理可以做什么。

**2、正向代理**

　　正向代理类似一个跳板机，代理访问外部资源。

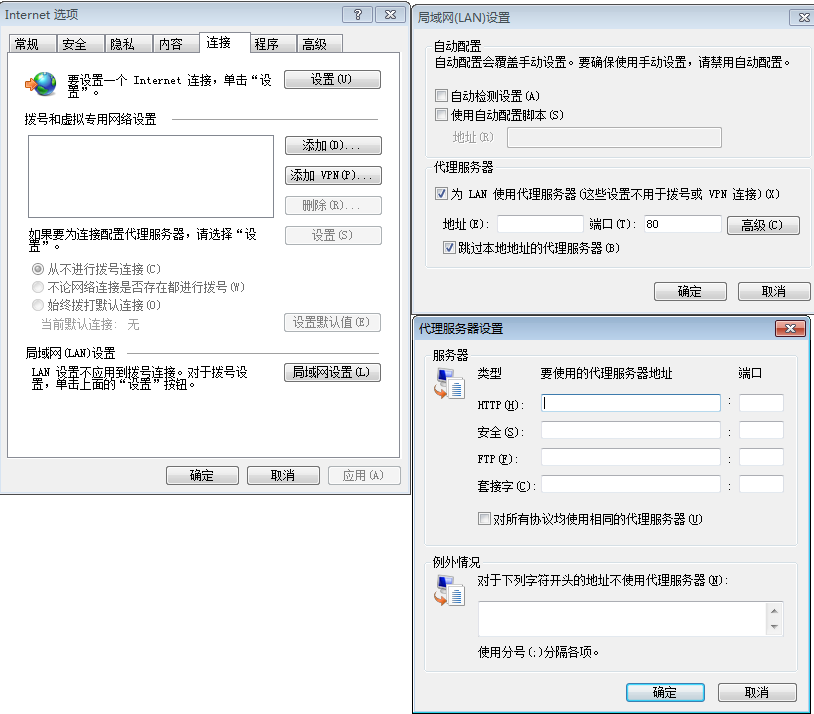


举个例子：

　　我是一个用户，我访问不了某网站，但是我能访问一个代理服务器，这个代理服务器呢,他能访问那个我不能访问的网站，于是我先连上代理服务器,告诉他我需要那个无法访问网站的内容，代理服务器去取回来,然后返回给我。从网站的角度，只在代理服务器来取内容的时候有一次记录，有时候并不知道是用户的请求，也隐藏了用户的资料，这取决于代理告不告诉网站。

**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。**

　　例如之前使用过这类软件例如CCproxy，[http://www.ccproxy.com](http://www.ccproxy.com/)/ 需要在浏览器中配置代理的地址。



　　总结来说：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。客户端必须要进行一些特别的设置才能使用正向代理。

**正向代理的用途：**

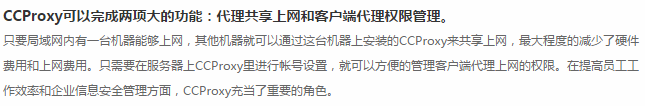
　（1）访问原来无法访问的资源，如google

     （2） 可以做缓存，加速访问资源

　　（3）对客户端访问授权，上网进行认证

　　（4）代理可以记录用户访问记录（上网行为管理），对外隐藏用户信息

例如CCProxy用途：



**3、反向代理**

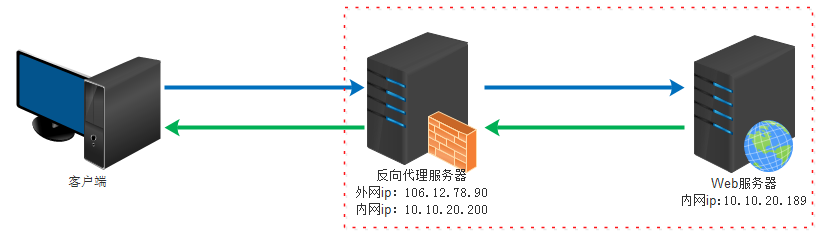
　　初次接触方向代理的感觉是，客户端是无感知代理的存在的，反向代理对外都是透明的，访问者者并不知道自己访问的是一个代理。因为客户端不需要任何配置就可以访问。

　　反向代理（Reverse Proxy）实际运行方式是指以代理服务器来接受internet上的连接请求，然后将请求转发给内部网络上的服务器，并将从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就表现为一个服务器。

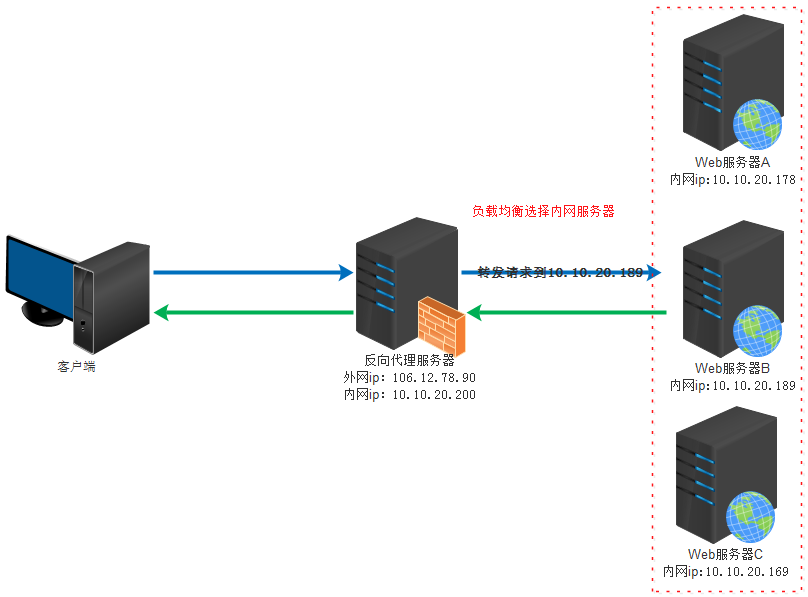
反向代理的作用：

（1）保证内网的安全，可以使用反向代理提供WAF功能，阻止web攻击

大型网站，通常将反向代理作为公网访问地址，Web服务器是内网。

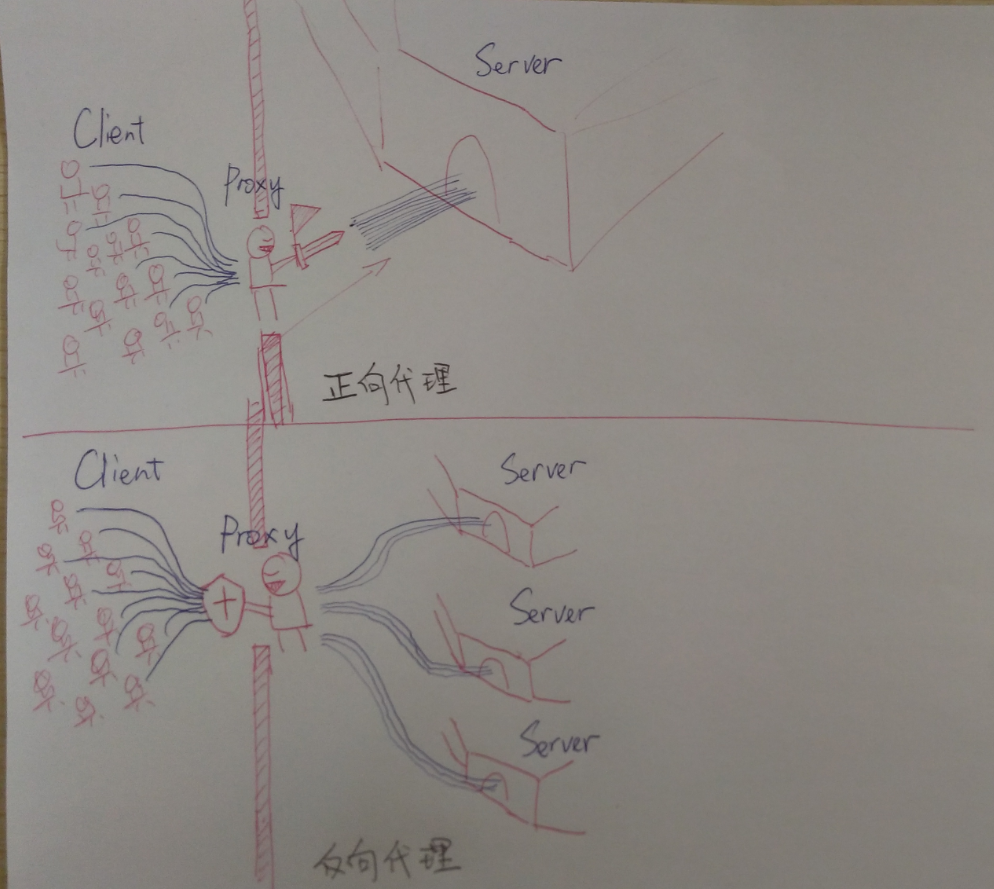


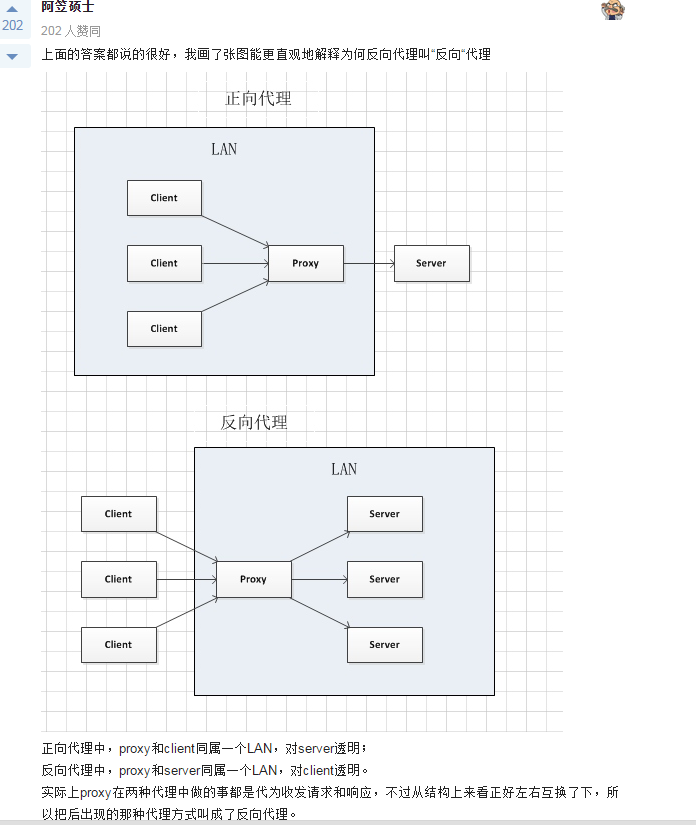
（2）负载均衡，通过反向代理服务器来优化网站的负载



**4、二者区别**

借用知乎两张图来表达：<https://www.zhihu.com/question/24723688>

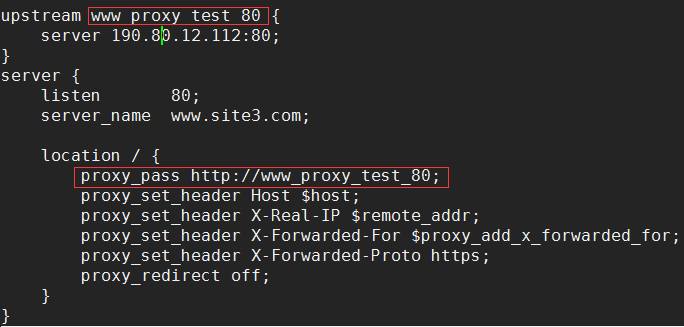




**5、nginx的反向代理**

 　　nginx支持配置反向代理，通过反向代理实现网站的负载均衡。这部分先写一个nginx的配置，后续需要深入研究nginx的代理模块和负载均衡模块。

nginx通过proxy\_pass\_http 配置代理站点，upstream实现负载均衡。



有区别，正向代理需要客户端配置，客户端是知道自己通过代理方式去访问服务器的。而反向代理，客户端根本就不知道。