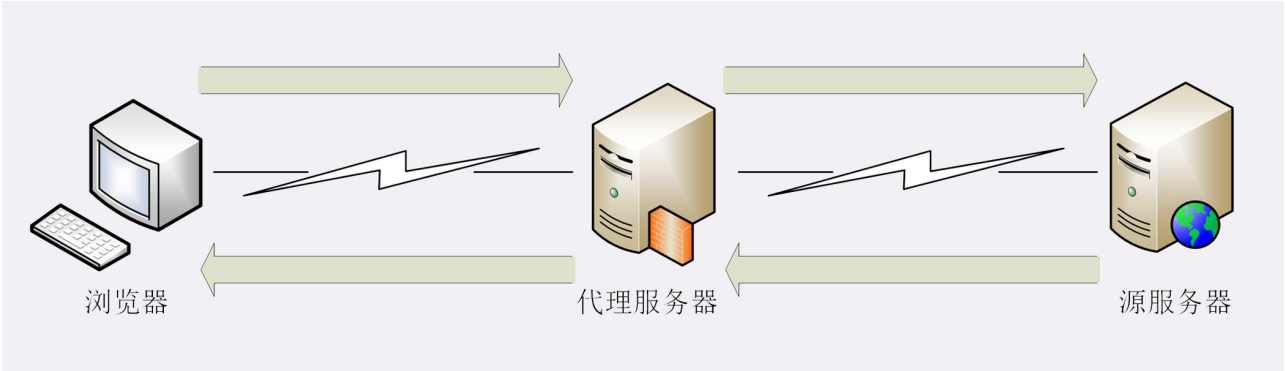
21-良心中间商：HTTP的代理服务

在前面讲HTTP协议的时候，我们严格遵循了HTTP的“请求-应答”模型，协议中只有两个互相通信的角色，分别是“请求方”浏览器（客户端）和“应答方”服务器。

今天，我们要在这个模型里引入一个新的角色，那就是HTTP代理。

引入HTTP代理后，原来简单的双方通信就变复杂了一些，加入了一个或者多个中间人，但整体上来看，还是一个有顺序关系的链条，而且链条里相邻的两个角色仍然是简单的一对一通信，不会出现越级的情况。



链条的起点还是客户端（也就是浏览器），中间的角色被称为代理服务器（proxy server），链条的终点被称为源服务器（origin server），意思是数据的“源头”“起源”。

代理服务

“代理”这个词听起来好像很神秘，有点“高大上”的感觉。

但其实HTTP协议里对它并没有什么特别的描述，它就是在客户端和服务器原本的通信链路中插入的一个中间环节，也是一台服务器，但提供的是“代理服务”。

所谓的“代理服务”就是指服务本身不生产内容，而是处于中间位置转发上下游的请求和响应，具有双重身份：面向下游的用户时，表现为服务器，代表源服务器响应客户端的请求；而面向上游的源服务器时，又表现为客户端，代表客户端发送请求。

还是拿上一讲的“生鲜超市”来打个比方。

之前你都是从超市里买东西，现在楼底下新开了一家24小时便利店，由超市直接供货，于是你就可以在便利店里买到原本必须去超市才能买到的商品。

这样超市就不直接和你打交道了，成了“源服务器”，便利店就成了超市的“代理服务器”。

在第4讲中，我曾经说过，代理有很多的种类，例如匿名代理、透明代理、正向代理和反向代理。

今天我主要讲的是实际工作中最常见的反向代理，它在传输链路中更靠近源服务器，为源服务器提供代理服务。

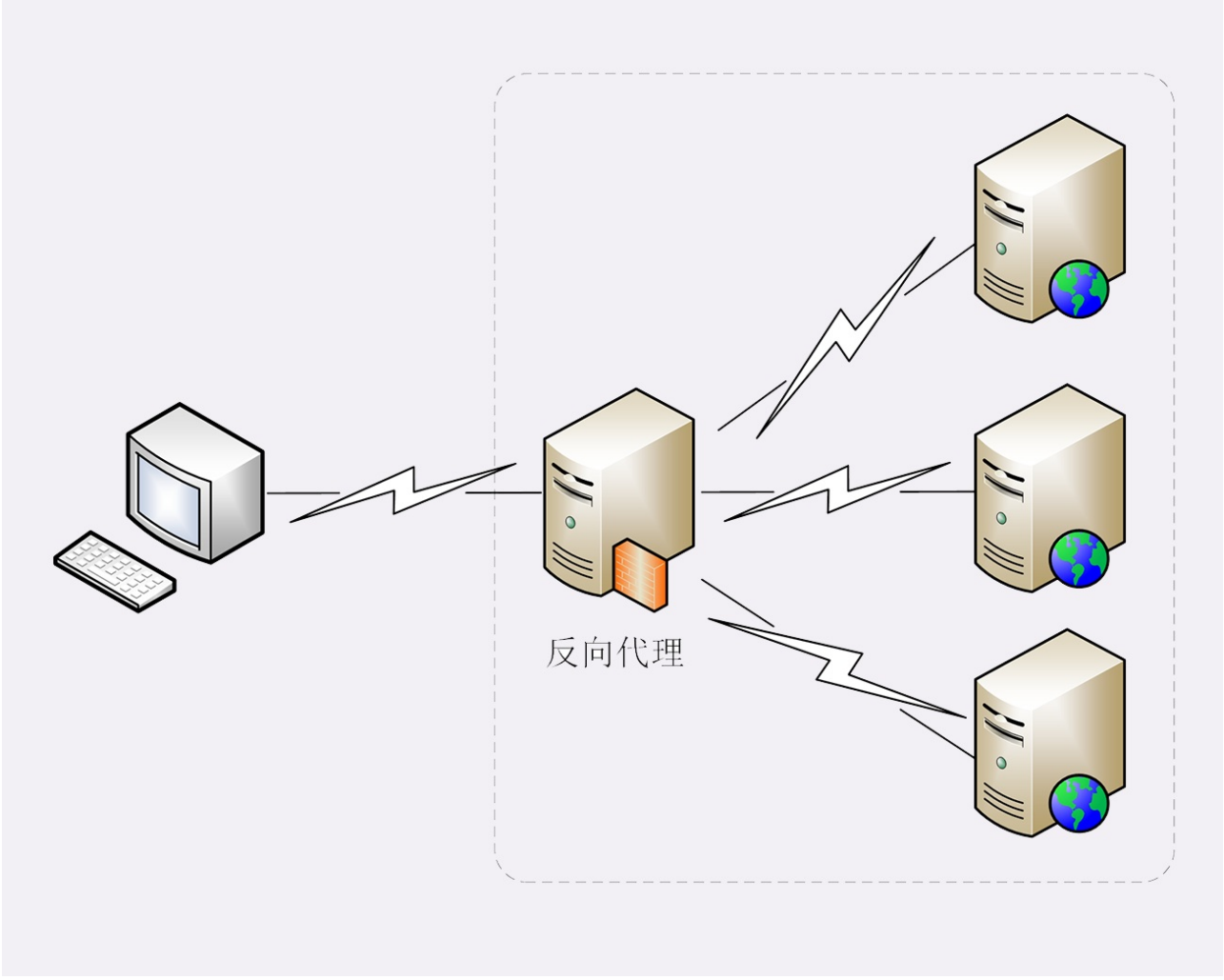
代理的作用

为什么要有代理呢？换句话说，代理能干什么、带来什么好处呢？

你也许听过这样一句至理名言：“计算机科学领域里的任何问题，都可以通过引入一个中间层来解决”（在这句话后面还可以再加上一句“如果一个中间层解决不了问题，那就再加一个中间层”）。TCP/IP协议栈是这样，而代理也是这样。

由于代理处在HTTP通信过程的中间位置，相应地就对上屏蔽了真实客户端，对下屏蔽了真实服务器，简单的说就是“**欺上瞒下**”。在这个中间层的“小天地”里就可以做很多的事情，为HTTP协议增加更多的灵活性，实现客户端和服务器的“双赢”。

代理最基本的一个功能是**负载均衡**。因为在面向客户端时屏蔽了源服务器，客户端看到的只是代理服务器，源服务器究竟有多少台、是哪些IP地址都不知道。于是代理服务器就可以掌握请求分发的“大权”，决定由后面的哪台服务器来响应请求。



代理中常用的负载均衡算法你应该也有所耳闻吧，比如轮询、一致性哈希等等，这些算法的目标都是尽量把外部的流量合理地分散到多台源服务器，提高系统的整体资源利用率和性能。

在负载均衡的同时，代理服务还可以执行更多的功能，比如：

- **健康检查**：使用“心跳”等机制监控后端服务器，发现有故障就及时“踢出”集群，保证服务高可用；
- **安全防护**：保护被代理的后端服务器，限制IP地址或流量，抵御网络攻击和过载；
- **加密卸载**：对外网使用SSL/TLS加密通信认证，而在安全的内网不加密，消除加解密成本；

- **数据过滤**：拦截上下行的数据，任意指定策略修改请求或者响应；
- **内容缓存**：暂存、复用服务器响应，这个与第20讲密切相关，我们稍后再说。

接着拿刚才的便利店来举例说明。

因为便利店和超市之间是专车配送，所以有了便利店，以后你买东西就更省事了，打电话给便利店让它去帮你取货，不用关心超市是否停业休息、是否人满为患，而且总能买到最新鲜的。

便利店同时也方便了超市，不用额外加大店面就可以增加客源和销量，货物集中装卸也节省了物流成本，由于便利店直接面对客户，所以也可以把恶意骚扰电话挡在外面。

代理相关头字段

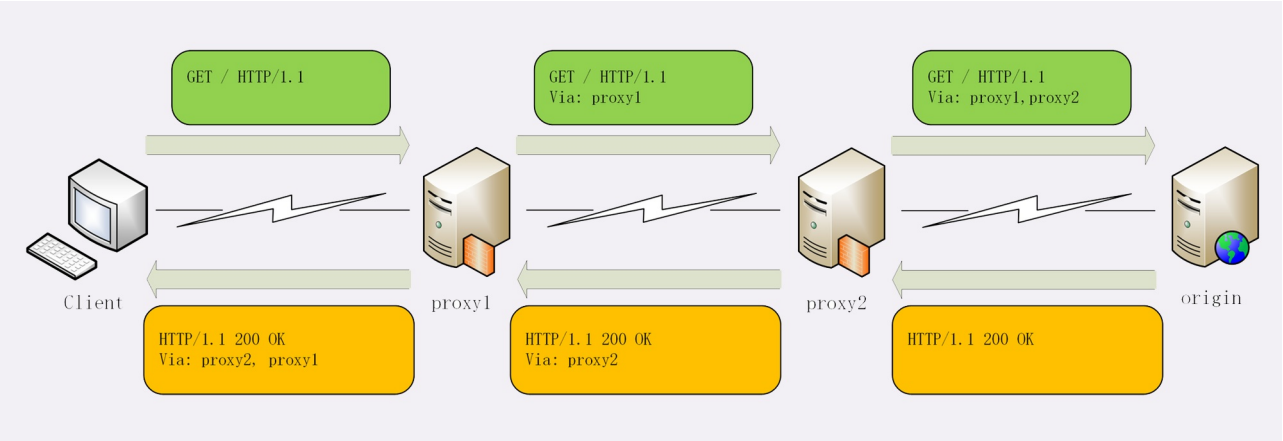
代理的好处很多，但因为它“欺上瞒下”的特点，隐藏了真实客户端和服务器，如果双方想要获得这些“丢失”的原始信息，该怎么办呢？

首先，代理服务器需要用字段“**Via**”标明代理的身份。

Via是一个通用字段，请求头或响应头里都可以出现。每当报文经过一个代理节点，代理服务器就会把自身的信息追加到字段的末尾，就像是经手人盖了一个章。

如果通信链路中有很多中间代理，就会在Via里形成一个链表，这样就可以知道报文究竟走过了多少个环节才到达了目的地。

例如下图中有两个代理：proxy1和proxy2，客户端发送请求会经过这两个代理，依次添加就是“Via: proxy1, proxy2”，等到服务器返回响应报文的时候就要反过来走，头字段就是“Via: proxy2, proxy1”。



Via字段只解决了客户端和源服务器判断是否存在代理的问题，还不能知道对方的真实信息。

但服务器的IP地址应该是保密的，关系到企业的内网安全，所以一般不会让客户端知道。不过反过来，通常服务器需要知道客户端的真实IP地址，方便做访问控制、用户画像、统计分析。

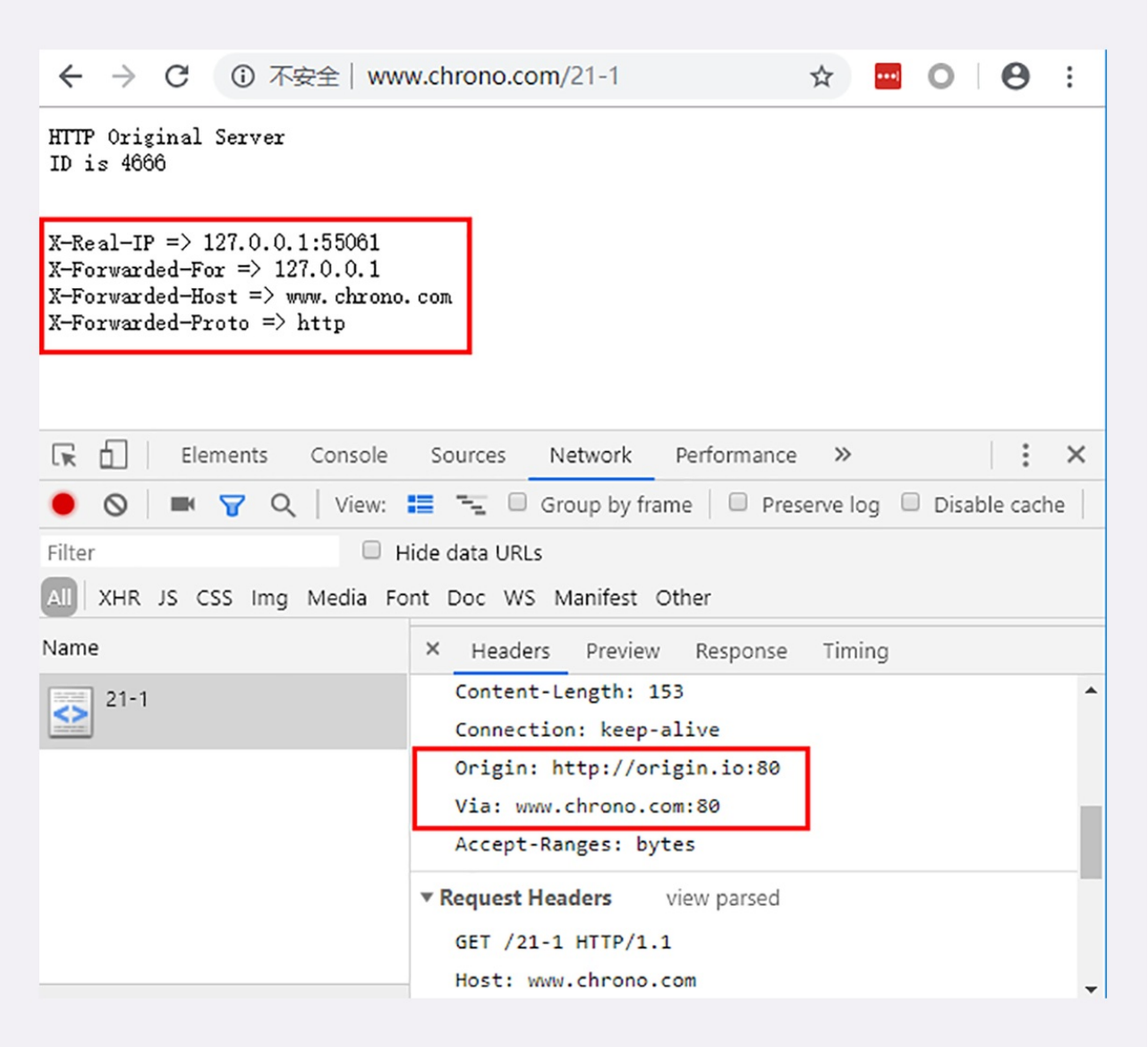
可惜的是HTTP标准里并没有为此定义头字段，但已经出现了很多“事实上的标准”，最常用的两个头字段是“**X-Forwarded-For**”和“**X-Real-IP**”。

“X-Forwarded-For”的字面意思是“为谁而转发”，形式上和“Via”差不多，也是每经过一个代理节点就

会在字段里追加一个信息。但“Via”追加的是代理主机名（或者域名），而“X-Forwarded-For”追加的是请求方的IP地址。所以，在字段里最左边的IP地址就客户端的地址。

“X-Real-IP”是另一种获取客户端真实IP的手段，它的作用很简单，就是记录客户端IP地址，没有中间的代理信息，相当于是“X-Forwarded-For”的简化版。如果客户端和源服务器之间只有一个代理，那么这两个字段的值就是相同的。

我们的实验环境实现了一个反向代理，访问“http://www.chrono.com/21-1”，它会转而访问“http://origin.io”。这里的“origin.io”就是源站，它会在响应报文里输出“Via”“X-Forwarded-For”等代理头字段信息：



单从浏览器的页面上很难看出代理做了哪些工作，因为代理的转发都在后台不可见，所以我把这个过程用Wireshark抓了一个包：

Protocol	Info
TCP	55061 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PERM=1
TCP	80 → 55061 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
TCP	55061 → 80 [ACK] Seq=1 Ack=1 Win=525568 Len=0
HTTP	GET /21-1 HTTP/1.1
TCP	80 → 55061 [ACK] Seq=1 Ack=446 Win=525568 Len=0
TCP	55063 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PERM=1
TCP	80 → 55063 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
TCP	55063 → 80 [ACK] Seq=1 Ack=1 Win=525568 Len=0
HTTP	GET /proxy/ HTTP/1.0
TCP	80 → 55063 [ACK] Seq=1 Ack=553 Win=525568 Len=0
HTTP	HTTP/1.1 200 OK (text/plain)
TCP	55063 → 80 [ACK] Seq=553 Ack=324 Win=525056 Len=0
TCP	80 → 55063 [FIN, ACK] Seq=324 Ack=553 Win=525568 Len=0
TCP	55063 → 80 [ACK] Seq=553 Ack=325 Win=525056 Len=0
TCP	55063 → 80 [FIN, ACK] Seq=553 Ack=325 Win=525056 Len=0
TCP	80 → 55063 [ACK] Seq=325 Ack=554 Win=525568 Len=0
HTTP	HTTP/1.1 200 OK (text/plain)
TCP	55061 → 80 [ACK] Seq=446 Ack=375 Win=525056 Len=0

从抓包里就可以清晰地看出代理与客户端、源服务器的通信过程：

1. 客户端55061先用三次握手连接到代理的80端口，然后发送GET请求；
2. 代理不直接生产内容，所以就代表客户端，用55063端口连接到源服务器，也是三次握手；
3. 代理成功连接源服务器后，发出了一个HTTP/1.0 的GET请求；
4. 因为HTTP/1.0默认是短连接，所以源服务器发送响应报文后立即用四次挥手关闭连接；
5. 代理拿到响应报文后再发回给客户端，完成了一次代理服务。

在这个实验中，你可以看到除了“X-Forwarded-For”和“X-Real-IP”，还出现了两个字段：“X-Forwarded-Host”和“X-Forwarded-Proto”，它们的作用与“X-Real-IP”类似，只记录客户端的信息，分别是客户端请求的原始域名和原始协议名。

代理协议

有了“X-Forwarded-For”等头字段，源服务器就可以拿到准确的客户端信息了。但对于代理服务器来说它并不是一个最佳的解决方案。

因为通过“X-Forwarded-For”操作代理信息必须要解析HTTP报文头，这对于代理来说成本比较高，原本只需要简单地转发消息就好，而现在却必须要费力解析数据再修改数据，会降低代理的转发性能。

另一个问题是“X-Forwarded-For”等头必须要修改原始报文，而有些情况下是不允许甚至不可能的（比如使用HTTPS通信被加密）。

所以就出现了一个专门的“代理协议”（The PROXY protocol），它由知名的代理软件HAProxy所定义，也是一个“事实标准”，被广泛采用（注意并不是RFC）。

“代理协议”有v1和v2两个版本，v1和HTTP差不多，也是明文，而v2是二进制格式。今天只介绍比较好理解的v1，它在HTTP报文前增加了一行ASCII码文本，相当于又多了一个头。

这一行文本其实非常简单，开头必须是“PROXY”五个大写字母，然后是“TCP4”或者“TCP6”，表示客户端的IP地址类型，再后面是请求方地址、应答方地址、请求方端口号、应答方端口号，最后用一个回车换行（\r\n）结束。

例如下面的这个例子，在GET请求行前多出了PROXY信息行，客户端的真实IP地址是“1.1.1.1”，端口号是55555。

```
PROXY TCP4 1.1.1.1 2.2.2.2 55555 80\r\n
GET / HTTP/1.1\r\n
Host: www.xxx.com\r\n
\r\n
```

服务器看到这样的报文，只要解析第一行就可以拿到客户端地址，不需要再去理会后面的HTTP数据，省了很多事情。

不过代理协议并不支持“X-Forwarded-For”的链式地址形式，所以拿到客户端地址后再如何处理就需要代理服务器与后端自行约定。

小结

1. HTTP代理就是客户端和服务器通信链路中的一个中间环节，为两端提供“代理服务”；
2. 代理处于中间层，为HTTP处理增加了更多的灵活性，可以实现负载均衡、安全防护、数据过滤等功能；
3. 代理服务器需要使用字段“Via”标记自己的身份，多个代理会形成一个列表；
4. 如果想要知道客户端的真实IP地址，可以使用字段“X-Forwarded-For”和“X-Real-IP”；
5. 专门的“代理协议”可以在不改动原始报文的情况下传递客户端的真实IP。

课下作业

1. 你觉得代理有什么缺点？实际应用时如何避免？
2. 你知道多少反向代理中使用的负载均衡算法？它们有什么优缺点？

欢迎你把自己的学习体会写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。

课外小贴士

01 现实生活中也有很多“代理”，例如房产代理、留学代理、保险代理、诉讼代理，可以对比理解一下。

02 知名的代理软件有 HAProxy、Squid、Varnish 等，而 Nginx 虽然是 Web 服务器，但也可以作为代理服务器，而且功能毫不逊色。

03 “Via”是 HTTP 协议里规定的标准头字段，但有的服务器返回的响应报文里会使用“X-Via”，含义是相同的。

04 因为 HTTP 是明文传输，请求头很容易被窜改，所以“X-Forwarded-For”也不是完全可信的。

05 RFC7239 定义了字段“Forwarded”，它可以代替“X-Forwarded-For”“X-Forwarded-Host”等字段，但应用得不多。

精选留言：

- -W.LI- 2019-07-15 08:34:28
  代理会增加链路长度，在代理上做一些复杂的处理。会很耗费性能，增加响应时间。
  1.随机
  2.轮询
  3.一致性hash
  4最近最少使用
  5.链接最少 [4赞]

  作者回复2019-07-15 09:22:46
  great！

- Geek_d4dee7 2019-07-16 07:25:57
  常听说的SLB是中间的这个代理么 老师

- Fstar 2019-07-15 21:54:02
  代理服务器如何连接源服务器？用 http1.0 短连接的效率不太好吧？集群一般都是局域网吗？

- 独步星空 2019-07-15 17:18:26
  老师，方便的时候能参照着http结构，顺带介绍下https么

  作者回复2019-07-15 17:45:46
  安全篇里全都是https，很快就要到了。

- 一步 2019-07-15 16:36:08
  对于代理协议，也有点疑问。代理协议只是在请求行的前面加了一个客服端地址和服务器地址，而没有整个请求的代理链路，如果源服务器想看整个请求经过了哪些代理怎么去看呢？(这个代理服务器没有再去修改X-Forwarded-For)

  作者回复2019-07-15 17:52:18
  代理协议设计的目的就不是记录传输链路，因为这个已经被X-Forwarded-For做了。

这样需求还是要用X-Forwarded-For字段。

- 一步 2019-07-15 16:23:25
  还出现了两个字段：“X-Forwarded-Host”和“X-Forwarded-Proto”，它们的作用与“X-Real-IP”类似，只记录客户端的信息，分别是客户端请求的原始域名和原始协议名。
  ——————————————
  老师，对于这句话，有点疑问，X-Forwarded-Host只是真实客户端的host吗？类比X-Real-IP，真实客户端的Host不应该是X-Real-Host吗？有关Forwarded的头不应该都是代理链路所有的以逗号分开的信息吗？

  作者回复2019-07-15 17:50:55
  是的，X-Forwarded-Host的格式与X-Forwarded-For不同，它只有一个值，不是逗号列表。

- Geek_54edc1 2019-07-15 10:37:46
  1、代理会成为性能瓶颈，有单点问题

  作者回复2019-07-15 11:48:17
  √

- lmingzhi 2019-07-15 07:44:04
  老师，请问有什么检测http代理ip匿名性的手段？

  是否只要检查请求头是否带有“X-Forwarded-For”和“X-Real-IP”及里面是否带有真实ip即可？

  作者回复2019-07-15 09:25:45
  如果代理比较“善良”，就会用“X-Forwarded-For”和“X-Real-IP”告知客户端的真实ip，如果它是完全匿名，不提供这些字段，我们也没有办法，因为它就是一个真实的客户端。