# 白话Nginx

## 一 Nginx是什么

Nginx是一个高性能的HTTP和反向代理服务器，也是一个 IMAP/POP3/SMTP 服务器。此篇，不会对nginx做过多的说明，网上的素材比比皆是，接下来主要对nginx两种代理方式（透明代理和反向代理），结合实践进行相应的描述。

首先，理解一下nginx的报文交互过程

如图1，分析nginx代理数据交互的流程：

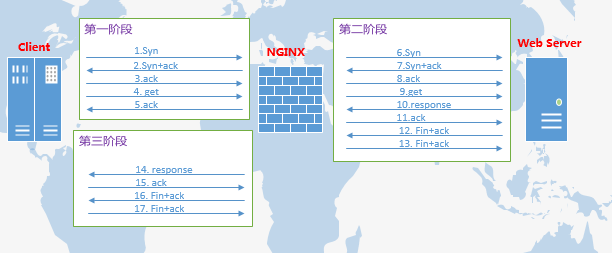


图1

1. 当client发起某一个URL的http请求时，首先将与nginx通过TCP的三次握手建立连接，连接成功后，client向nginx发起get请求
2. Nginx代理client的请求与web server建立TCP连接，并且向web server发起get请求，web server接收到get请求后，回复response给nginx，nginx成功接收到响应后向server发起ack确认，server接收到ack后断开与nginx的连接
3. Nginx将web server的response响应发送给client，并且nginx接收到client ack报文后与客户端断开连接

从整个数据交互过程来看，http请求建立在web server短连接的基础上，其中nginx与client和server建立tcp双向的连接，对client而言nginx为web server，对web server而言，nginx就是客户端，整个过程，nginx扮演着数据劫持者的角色，并且很好的隐藏了自己的身份。

## 二Nginx透明代理

透明代理的原理是客户端根本不需要知道有代理服务器的存在，nginx改变request fields(报文)，以nginx IP为源IP向server发起请求，并且nginx接收server的响应后，再转发给客户端，结合这种实现方式，安全网关利用nginx透明代理，实现了上层的数据应用处理，从而完成对数据报文的检测或者内容替换等工作。



图

如图2，数据平台采用nginx透明代理的模式，通常这种部署模式为：Br+iptables+nginx，其原理，是把整个网关平台串行在企业网络里，网关平台部署到核心交换上行，企业内部客户端发起的HTTP/HTTPS请求，默认会经过nginx proxy，经由网关代理客户端发起的http请求与server进行通信。

分析这种部署发现，需要nginx服务器具有高吞吐的处理能力，以及异常情况的响应能力，最主要不能影响企业内部用户的上网体验，其次对于https协议网站，为了避免用户端浏览时弹出网站不信任页签，动态证书签发机制（部署证书伪签发机制）是一个不可缺少的模块。

首先分析一下所谓的Br+iptables+nginx：

1）Br：linux内核是通过一个虚拟的网桥设备来实现桥接的。网桥设备br0绑定了eth0和eth1。对于网络协议栈的上层来说，只看得到br0，因为桥接是在数据链路层实现，上层不需要关心桥接的细节。于是协议栈上层需要发送的报文被送到br0，网桥设备的处理程序再来判断报文该被转发到eth0或是eth1，或者两者皆是；反过来，从eth0或从eth1接收到的报文被提交给网桥的处理程序，在这里会判断报文该转发、丢弃、或提交到协议栈上层进行处理。

2）Iptables：用于监听tcp 80/443端口的数据并重定向HTTP数据到nginx，nginx对http请求进行相应的处理后，代替客户端向server端发起请求，但这里要提到一个问题：iptables nf\_conntrack会话跟踪表是有限制的，会在内核空间(内存)中记录所追踪的连接的状态，通过对nginx的评估测试发现，会遇到跟踪表满的情况以及连接并发的瓶颈，从而影响了业务数据的转发能力。

图

如图3所示，当数据进入linux内核处理流程时，数据报文缓存在接收队列，报文由接收队列进行上送，首先经过第一个hook点pre-routing，此时根据设置的iptables nat表，把基于tcp协议的访问端口为80/443的报文进行重定向到本地处理，通过对http数据payload处理后，再进行协议封装转发出去。而未过滤的流量经过Br直接转发，不对报文做任何的处理。

通过实际测试后得到数据显示，企业端面对大量的HTTP/HTTPS的请求，往往对性能的要求很高，尤其要面对企业大数据的挑战， 从而nginx代理压力会很大，代理风险会很高，面对企业自有的代理服务器，也会屏蔽掉nginx单一IP的大量请求，其次再经过上层应用数据处理后，稳定性很难保证。

总而言之，nginx透明代理的实现，很难保证企业数据交互的稳定性，更适用于小型企业。

## 三Nginx反向代理

反向代理的原理是指以代理服务来接收Internet上的连接请求，然后将请求转发给网络上的服务器，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反向代理服务器。网络设备利用nginx反向代理的方式，企业内部由DNS server将业务流量牵引到nginx服务器，从而实现对客户请求的反向代理。



图

反向代理的部署方式，如图4，其部署简单，无需改变用户的网络结构，用户可以通过配置DNS server，强制指定客户端访问saas应用的URL对应的IP为nginx Gateway，这样一来流量就会被牵引到网关设备上，再由nginx向saas应用发起相应的http请求。

支持基于Saas云服务，相比透明代理其具有以下优势：性能更优越，由于反向代理不使用DNS过滤，iptables重定向和nginx的动态证书功能，这使得nginx并发处理性能有很大幅度的提升，saas业务之外的数据不需要经过nginx网关，从而解决了数据压力的难题。



图

如图5，当数据进入linux内核处理流程时，数据报文缓存到接收队列，报文由收队列进行上送，首先经过第一个hook点pre-routing，此时根据检查报文中目的IP为本地后，经过local\_in上送到四层及以上协议栈进行数据的处理，通过对数据payload处理后，再进行协议封装，根据路由的选择将报文转发出去。

经过测试分析，为了解决http并发限制的问题，可以通过配置nginx.conf文件，设置多个nginx的代理IP，来对server端发起请求，从而解决并发受限的问题，反向代理的实现，解决了厂商面对的代理性能瓶颈的难题，提升了代理服务的业务能力，从而实现了企业高性能的代理部署。

综上， 通过介绍nginx的透明/反向代理的实现，网络安全设备使用Nginx的场景比较广泛单纯代理而言，nginx很优秀，如果结合上层应用处理，性能表现需要结合场景应用分析，面对大数据的挑战，nginx的实践与应用，还需要进一步研究，说到底一句话“路漫漫其修远兮，吾将上下而求索”~~~