# Http知识地图：第一篇重新看

地图上有字

描述已自动生成

# 万维网(WWW:world wide web)

万维网WWW是互联网Internet的一部分。

1989 年，任职于欧洲核子研究中心（CERN）的蒂姆·伯纳斯 - 李（Tim Berners-Lee）发表了一篇论文，提出了在互联网上构建超链接文档系统的构想。这篇论文中他确立了三项关键技术。

URI：即统一资源标识符，作为互联网上资源的唯一身份；

HTML：即超文本标记语言，描述超文本文档；

HTTP：即超文本传输协议，用来传输超文本。

这三项技术在如今的我们看来已经是稀松平常，但在当时却是了不得的大发明。基于它们，就可以把超文本系统完美地运行在互联网上，让各地的人们能够自由地共享信息，蒂姆把这个系统称为“万维网”（World Wide Web），也就是我们现在所熟知的 Web。

# 二、HTTP的发展历程

1.1989年，HTTP 协议始于三十年前蒂姆·伯纳斯 - 李的一篇论文；

2.20世纪90年代初期，HTTP/0.9 是个简单的文本协议，只能获取文本资源；

3.1993年，HTTP/1.0 确立了大部分现在使用的技术，但它不是正式标准；

4.1995年，HTTP/1.1 是目前互联网上使用最广泛的协议，功能也非常完善；它是一个“正式的标准”，而不是一份可有可无的“参考文档”。经典版本。

5.2015年，HTTP/2 基于 Google 的 SPDY 协议，注重性能改善，但还未普及；主要特点：

1).二进制协议，不再是纯文本；

2).可发起多个请求，废弃了 1.1 里的管道；

3).使用专用算法压缩头部，减少数据传输量；

4).允许服务器主动向客户端推送数据；

5).增强了安全性，“事实上”要求加密通信。

6.2018年，HTTP/3 基于 Google 的 QUIC 协议，是将来的发展方向。

2018 年，互联网标准化组织 IETF 提议将“HTTP over QUIC”更名为“HTTP/3”并获得批准，HTTP/3 正式进入了标准化制订阶段，也许两三年后就会正式发布，到时候我们很可能会跳过 HTTP/2 直接进入 HTTP/3。

# 三、http是什么

HTTP 就是超文本传输协议，也就是 HyperText Transfer Protocol。延伸：

1,你是怎么理解 HTTP 字面上的“超文本”和“传输协议”的？

2.能否谈一下你对 HTTP 的认识？越多越好。

3.HTTP 有什么特点？有什么优点和缺点？

4.HTTP 下层都有哪些协议？是如何工作的？

超文本：它是文字、图片、音频和视频等的混合体，最关键的是含有“超链接”，能够从一个“超文本”跳跃到另一个“超文本”，形成复杂的非线性、网状的结构关系。

传输：就是把一堆东西从 A 点搬到 B 点，或者从 B 点搬到 A 点，即“A<===>B”，双向传输。

协议：大家都认同的一种行为约定与规范。

总结：

1.HTTP 是一个用在计算机世界里的协议，它确立了一种计算机之间交流通信的规范，以及相关的各种控制和错误处理方式。

2.HTTP 专门用来在两点之间传输数据，不能用于广播、寻址或路由。

3.HTTP 传输的是文字、图片、音频、视频等超文本数据。

4.HTTP 是构建互联网的重要基础技术，它没有实体，依赖许多其他的技术来实现，但同时许多技术也都依赖于它。

在互联网世界里，HTTP 通常跑在 TCP/IP 协议栈之上，依靠 IP 协议实现寻址和路由、TCP 协议实现可靠数据传输、DNS 协议实现域名查找、SSL/TLS 协议实现安全通信。此外，还有一些协议依赖于 HTTP，例如 WebSocket、HTTPDNS 等。这些协议相互交织，构成了一个协议网，而 HTTP 则处于中心地位。

# 四、浏览器是什么

浏览器的正式名字叫“Web Browser”，顾名思义，就是检索、查看互联网上网页资源的应用程序，名字里的 Web，实际上指的就是“World Wide Web”，也就是万维网。

浏览器本质上是一个 HTTP 协议中的请求方，使用 HTTP 协议获取网络上的各种资源。当然，为了让我们更好地检索查看网页，它还集成了很多额外的功能。例如，HTML 排版引擎用来展示页面，JavaScript 引擎用来实现动态化效果，甚至还有开发者工具用来调试网页，以及五花八门的各种插件和扩展。

在 HTTP 协议里，浏览器的角色被称为“User Agent”即“用户代理”，意思是作为访问者的“代理”来发起 HTTP 请求。不过在不引起混淆的情况下，我们通常都简单地称之为“客户端”。

# 五、服务器是什么

刚才说的浏览器是 HTTP 里的请求方，那么在协议另一端的应答方（响应方）又是什么呢？这个你一定也很熟悉，答案就是服务器，Web Server。

Web 服务器是一个很大也很重要的概念，它是 HTTP 协议里响应请求的主体，通常也把控着绝大多数的网络资源，在网络世界里处于强势地位。当我们谈到“Web 服务器”时有两个层面的含义：硬件和软件。

硬件含义就是物理形式或“云”形式的机器，在大多数情况下它可能不是一台服务器，而是利用反向代理、负载均衡等技术组成的庞大集群。但从外界看来，它仍然表现为一台机器，但这个形象是“虚拟的”。

软件含义的 Web 服务器可能我们更为关心，它就是提供 Web 服务的应用程序，通常会运行在硬件含义的服务器上。它利用强大的硬件能力响应海量的客户端 HTTP 请求，处理磁盘上的网页、图片等静态文件，或者把请求转发给后面的 Tomcat、Node.js 等业务应用，返回动态的信息。

比起层出不穷的各种 Web 浏览器，Web 服务器就要少很多了，一只手的手指头就可以数得过来。

Apache 是老牌的服务器，到今天已经快 25 年了，功能相当完善，相关的资料很多，学习门槛低，是许多创业者建站的入门产品。

Nginx 是 Web 服务器里的后起之秀，特点是高性能、高稳定，且易于扩展。自 2004 年推出后就不断蚕食 Apache 的市场份额，在高流量的网站里更是不二之选。

此外，还有 Windows 上的 IIS、Java 的 Jetty/Tomcat 等，因为性能不是很高，所以在互联网上应用得较少。

六、CDN是什么

浏览器和服务器是 HTTP 协议的两个端点，那么，在这两者之间还有别的什么东西吗？当然有了。

浏览器通常不会直接连到服务器，中间会经过“重重关卡”，其中的一个重要角色就叫做 CDN。

CDN，全称是“Content Delivery Network”，翻译过来就是“内容分发网络”。

它应用了 HTTP 协议里的缓存和代理技术，代替源站响应客户端的请求。CDN 有什么好处呢？简单来说，它可以缓存源站的数据，让浏览器的请求不用“千里迢迢”地到达源站服务器，直接在“半路”就可以获取响应。如果 CDN 的调度算法很优秀，更可以找到离用户最近的节点，大幅度缩短响应时间。打个比方，就好像唐僧西天取经，刚出长安城，就看到阿难与迦叶把佛祖的真经递过来了，是不是很省事？

CDN 也是现在互联网中的一项重要基础设施，除了基本的网络加速外，还提供负载均衡、安全防护、边缘计算、跨运营商网络等功能，能够成倍地“放大”源站服务器的服务能力，很多云服务商都把 CDN 作为产品的一部分，我也会在后面用一讲的篇幅来专门讲解 CDN。

六、爬虫

前面说到过浏览器，它是一种用户代理，代替我们访问互联网。但 HTTP 协议并没有规定用户代理后面必须是“真正的人类”，它也完全可以是“机器人”，这些“机器人”的正式名称就叫做“爬虫”（Crawler），实际上是一种可以自动访问 Web 资源的应用程序。

“爬虫”这个名字非常形象，它们就像是一只只不知疲倦的、辛勤的蚂蚁，在无边无际的网络上爬来爬去，不停地在网站间奔走，搜集抓取各种信息。据估计，互联网上至少有 50% 的流量都是由爬虫产生的，某些特定领域的比例还会更高，也就是说，如果你的网站今天的访问量是十万，那么里面至少有五六万是爬虫机器人，而不是真实的用户。

爬虫是怎么来的呢？绝大多数是由各大搜索引擎“放”出来的，抓取网页存入庞大的数据库，再建立关键字索引，这样我们才能够在搜索引擎中快速地搜索到互联网角落里的页面。

爬虫也有不好的一面，它会过度消耗网络资源，占用服务器和带宽，影响网站对真实数据的分析，甚至导致敏感信息泄漏。所以，又出现了“反爬虫”技术，通过各种手段来限制爬虫。其中一项就是“君子协定”robots.txt，约定哪些该爬，哪些不该爬。无论是“爬虫”还是“反爬虫”，用到的基本技术都是两个，一个是 HTTP，另一个就是 HTML。

反爬虫：验证你不是机器人。

# 七、HTML/WebService/WAF

到现在我已经说完了图中右边的五大部分，而左边的 HTML、WebService、WAF 等由于与 HTTP 技术上实质关联不太大，所以就简略地介绍一下，不再过多展开。

HTML 是 HTTP 协议传输的主要内容之一，它描述了超文本页面，用各种“标签”定义文字、图片等资源和排版布局，最终由浏览器“渲染”出可视化页面。HTML 目前有两个主要的标准，HTML4 和 HTML5。广义上的 HTML 通常是指 HTML、JavaScript、CSS 等前端技术的组合，能够实现比传统静态页面更丰富的动态页面。

接下来是 Web Service，它的名字与 Web Server 很像，但却是一个完全不同的东西。Web Service 是一种由 W3C 定义的应用服务开发规范，使用 client-server 主从架构，通常使用 WSDL 定义服务接口，使用 HTTP 协议传输 XML 或 SOAP 消息，也就是说，它是一个基于 Web（HTTP）的服务架构技术，既可以运行在内网，也可以在适当保护后运行在外网。因为采用了 HTTP 协议传输数据，所以在 Web Service 架构里服务器和客户端可以采用不同的操作系统或编程语言开发。例如服务器端用 Linux+Java，客户端用 Windows+C#，具有跨平台跨语言的优点。

WAF 是近几年比较“火”的一个词，意思是“网络应用防火墙”。与硬件“防火墙”类似，它是应用层面的“防火墙”，专门检测 HTTP 流量，是防护 Web 应用的安全技术。WAF 通常位于 Web 服务器之前，可以阻止如 SQL 注入、跨站脚本等攻击，目前应用较多的一个开源项目是 ModSecurity，它能够完全集成进 Apache 或 Nginx。

小结：

互联网上绝大部分资源都使用 HTTP 协议传输；

浏览器是 HTTP 协议里的请求方，即 User Agent；

服务器是 HTTP 协议里的应答方，常用的有 Apache 和 Nginx；

CDN 位于浏览器和服务器之间，主要起到缓存加速的作用；

爬虫是另一类 User Agent，是自动访问网络资源的程序。

提问：

你觉得 CDN 在对待浏览器和爬虫时会有差异吗？为什么？

你怎么理解 WebService 与 Web Server 这两个非常相似的词？

CDN在对待浏览器和爬虫时没有差异，因为如果没有验证码或者其他验证方式区分的话，浏览器和爬虫都被视为User Agent（客户代理）。 CDN 应当是不区分的，因为爬虫本身也是对 Web 资源的访问，且对于爬虫识别并不是 100% 准确的，因此 CDN 只会去计算实际使用了多少资源而不管其中多少来自爬虫。

Webservice是服务，Web Server是服务器。Web Service 是网络服务实体，而 Web Server 是网络服务器，后者的存在是为了承载前者。

# 二十、键入网址再按下回车，后面究竟发生了什么？