1. Web请求过程
2. BS架构概述

BS架构采用HTTP协议来交互数据，无状态的短连接能够快速地响应更多的用户。当用户输入URL时，首先请求DNS将这个域名解析为对应的IP地址，然后根据IP地址找到对应的服务器，发起get请求，由这个服务器决定返回数据给访问的用户。当数据返回给用户后，浏览器解析数据发现还有一些静态资源，又会发起新的HTTP请求，这些请求可能会在CDN上，那么CDN服务器又会处理这个用户的请求。

1. 如何发起一个请求

发起一个HTTP请求，首先浏览器要在建立Socket连接之前根据URL的域名解析出IP地址，再根据这个IP地址和默认的80端口与远程服务器建立Socket连接，然后浏览器根据这个URL地址组装成一个get类型的HTTP请求，通过outputStrem.write发送给目标服务器，服务器等待inputStream.read返回数据，最后断开这个连接。

如HttpClient包就可以发送HTTP请求，Linux中的curl命令也可以发起HTTP请求。

1. HTTP解析

要理解HTTP，既要熟悉HTTP Header，它控制着用户数据的传输，用户浏览器的渲染行为和服务器的执行逻辑，HTTP请求头会将客户端的详细信息以及请求的信息告诉服务器，而HTTP响应头会将响应信息返回给客户端。

1. 浏览器缓存

浏览器自身会对数据进行缓存，如果我们要直接获取最新的数据，则使用CTRL+F5来重新请求该页面，因为此时浏览器会重新向目标URL发送请求，不会使用浏览器缓存的数据，即使请求到达服务端，也可能访问到缓存的数据，因为服务端的前端部分也可能会设置一个缓存服务器来限流，为了获取最新的数据，需要通过HTTP来控制。在HTTP请求头中会添加一些请求头，如Cache-Control:no-cache，以及Pragma:no-cache，这样服务端就会清楚需要返回最新的数据，而不是缓存。

1. DNS域名解析

DNS解析过程如下：

1. 浏览器检查缓存中是否有该域名对应的IP地址，缓存是有TTL和大小限制的，IP的缓存不宜过长或过短，过长会导致不能及时检测到IP的变化，过短消耗请求性能；
2. 浏览器查找操作系统缓存中是否有该域名对应的IP地址，如hosts文件中，你可以将任何域名解析到任何能够访问的IP地址上，这种本地DNS很可能会被黑客利用，修改域名解析就能将特定的域名劫持到它指定的IP地址上，这就是域名劫持；
3. 如果本地都找不到对应的IP地址，则会向我们配置的DNS服务器地址中发送请求，系统会将域名发送给设置的LDNS（本地的域名服务器），它们会缓存域名解析结果，LDNS承担了80%的域名解析工作；
4. 如果LDNS没有命中，则直接到Root Server域名服务器请求解析；
5. Root Server返回给LDNS一个所查询域的主域名服务器（gTLD Server）地址，gTLD是国际顶级域名服务器（.com, .cn, .org）；
6. LDNS再向gTLD服务器发送请求；
7. gTLD服务器查找并返回此域名对应的Name Server域名服务器的地址，这个Name Server服务器就是你注册的域名服务器，如taobao.com就是Name Server服务器，也可能会有多级的Name Server；
8. Name Server域名服务器会查询存储的域名和IP映射表，将查询得到的IP和一个TTL值返回给DNS Server域名服务器；
9. LDNS会缓存这个域名和IP对应的关系，缓存时间由TTL控制；
10. 将解析的结果返回给用户，用户根据TTL值缓存在本地系统的缓存中，域名解析结束；

跟踪域名解析过程可以通过nslookup命令来查询域名解析过程，DNS域名解析会在LDNS和本地机器上缓存，LDNS上的缓存只能通过TTL到期来删除，而本地机器的缓存可以通过ipconfig /flushdns来刷新缓存，Linux中通过 /etc/init.d/nscd restart来清除缓存。

1. CDN工作机制
2. 概述

CDN是内容分布网络，它是一种先进的流量分配网络，通过在现有的Internet中增加一层新的网络架构，将网站的内容发布到最接近用户的网络边缘，使用户可以就近取得数据，提高响应速度。CDN的功能是镜像+缓存+负载均衡，目前CDN用于缓存静态资源为主，用户在主站服务器请求到动态资源后，再从CDN中下载静态数据。

1. CDN访问过程

一个用户访问某个静态文件，首先向LDNS发起请求，经过迭代解析后回到这个域名的NameServer服务器去解析，通过DNS解析服务器解析到另一个域名，该域名最终会被指向CDN全局中的DNS负载均衡服务器，再由这个服务器分配是哪个地方的访问用户，返回给离这个用户最近的CDN节点，用户直接去这个CDN节点获取文件，如果文件不存在，则回到源站去获取文件。

1. CDN动态加速

在CDN的DNS解析过程中通过动态的链路探测来寻找回源的最好路径，通过DNS的调度将所有请求调度到选定的这条路径上回源，从而加速用户访问速度。CDN节点是遍布全国的，当用户接入一个CDN节点后，可以选择一条从离用户最近的CDN节点到源站链路最好的路径让用户走，怎么知道哪条路径最好呢？在每个CDN节点上从源站上下载一定大小的文件，看看哪个链路耗时最短，这样构成的链路链表绑定到DNS解析上，更新到CDN的LDNS中。