

DNS解析

域名系统 (DNS) 的工作

简单来说，就像一个巨大的电话簿，它把你输入的网址（比如`www.google.com`）转换成浏览器可以理解的、用于定位服务器的IP地址（比如`142.250.199.110`）。

对于前端开发者来说，理解DNS解析流程非常重要，因为它直接影响你网站的加载速度。

DNS域名解析流程

1. 浏览器缓存与本地缓存

当你输入一个网址时，浏览器会首先检查自己的**浏览器缓存**中是否有这个域名的IP地址。如果有，就会直接使用，解析结束。

如果浏览器缓存中没有，它会检查操作系统的**本地缓存**（`/etc/hosts`文件）。如果这里有记录，同样会直接使用。

2. 本地DNS服务器（递归解析）

如果以上缓存都没有命中，浏览器就会向你配置的**本地DNS服务器**（通常是你的网络服务商ISP提供的，比如电信、联通，或者你手动配置的`8.8.8.8`）

发起一个**递归查询**请求。这个请求的意思是：“请你帮我找到`www.google.com`的IP地址，并且把最终结果返回给我。”

本地DNS服务器收到请求后，它会承担起寻找IP地址的全部责任。这就是**递归解析**。

3. 权威解析

本地DNS服务器会为了你的请求，开始一系列的**权威解析**（或迭代查询）。这个过程就像一个寻宝游戏：

根DNS服务器：本地DNS服务器首先会询问**根DNS服务器**（全球有13个，由顶级域名机构运营）：“请问`.com`这个域名的地址在哪？”

根DNS服务器会告诉它：“我不知道www.google.com的IP地址，但是我知道.com这个**顶级域名服务器**的地址。”

顶级域名（TLD）服务器：接下来，本地DNS服务器会去询问**顶级域名服务器**（例如.com、.cn、.org等）：“请问google.com这个域名的地址在哪？”

TLD服务器会告诉它：“我不知道www.google.com的IP地址，但是我知道google.com这个**权威DNS服务器**的地址。”

权威DNS服务器：最后，本地DNS服务器会去询问**权威DNS服务器**（这个域名由其所有者，比如Google公司自己维护）：“请问www.google.com的IP地址是什么？”

权威DNS服务器会告诉它：“www.google.com的IP地址是142.250.199.110。”

4.返回IP地址与DNS缓存

当本地DNS服务器拿到最终的IP地址后，它会：

- 把这个IP地址返回给你的浏览器。
- 将这个IP地址**缓存**起来，以备下次有人请求相同的域名时，可以直接返回，无需重复上述复杂的查询过程。缓存的有效时间由TTL（Time To Live）值决定。

DNS缓存

理解**DNS缓存**对于提升网站性能至关重要。

- **浏览器缓存：**浏览器会缓存它**最近访问过的域名的IP地址**。
- **本地DNS服务器缓存：**本地DNS服务器会缓存它**查询过的域名IP地址**。
- **操作系统缓存：**操作系统也会缓存DNS记录，通常是**存在hosts文件中**。

缓存是DNS系统高效运行的关键。如果每次访问网站都要从根DNS服务器开始查询，**那网站的加载速度会非常慢**。

DNS预解析

DNS预解析（DNS prefetching）是前端性能**优化的一种技术**。

它的原理是在用户**点击链接或访问资源之前**，**浏览器就提前进行DNS解析**。这样当用户真正需要访问这些资源时，解析过程已经完成，**节省了宝贵的等待时间**。

你可以在HTML的<head>标签中使用以下代码来启用DNS预解析：

```
<link rel="dns-prefetch" href="https://example.com">
```

这告诉浏览器，**你可能需要访问example.com上的资源**

请提前解析其DNS。这个技术对于那些需要加载来自不同域名的外部资源（比如字体、CDN上的图片、第三方脚本等）的网站特别有用。

DNS预连接

预连接（Preconnect）是前端性能优化中的一个重要技术，它和前面提到的 DNS 预解析很像，但功能更强大。

如果说 DNS 预解析只是提前把“电话号码”查出来，**那么预连接就是提前把“电话”打通，但暂时不说话。**

预连接是一种浏览器优化技术，它告诉浏览器，**你即将需要从某个域名下载资源**。浏览器收到这个指令后，就会在用户真正发起请求之前，**提前完成与这个域名服务器的连接建立过程。**

这个连接建立过程通常包括以下几个步骤：

- 1. DNS 解析：**查找域名的 IP 地址。
- 2. TCP 握手：**客户端和服务器之间建立 TCP 连接。
- 3. TLS/SSL 协商：**如果使用了 HTTPS，还需要进行加密连接的建立。

这些步骤加起来会消耗不少时间，尤其是在网络条件不佳的情况下。预连接就是把这部分耗时操作提前完成

当真正需要请求资源时，**可以直接发送 HTTP 请求，节省了宝贵的等待时间。**

预连接的应用场景

预连接特别适用于那些你知道但又不能确定何时会需要的关键外部资源。常见的应用场景包括：

- **加载来自 CDN 的静态资源：**比如图片、字体、CSS 和 JavaScript 文件。
- **使用第三方 API：**比如社交媒体分享按钮、地图服务或支付服务。
- **加载 Web 字体：**很多网站会从 Google Fonts 或 Typekit 等外部服务加载字体。

如何使用预连接？

你可以在 HTML 的 `<head>` 标签中使用 `<link>` 标签来启用预连接。它的语法非常简单：

HTML

```
<link rel="preconnect" href="https://example.com">
```

这里的 href 指向你想要预连接的域名。

预连接 vs. DNS 预解析

预连接和 DNS 预解析都是性能优化技术，但它们之间有明确的区别：

- **DNS 预解析 (dns-prefetch)：**只做 DNS 查询。它的开销很小，可以对多个域名使用，是相对保守的优化手段。
-
- **预连接 (preconnect)：**除了 DNS 查询，还完成了 TCP 握手和 TLS/SSL 协商。它的开销更大，因为它实际上已经建立了连接。因此，不建议对太多域名使用，通常只对最关键的几个外部域名使用。