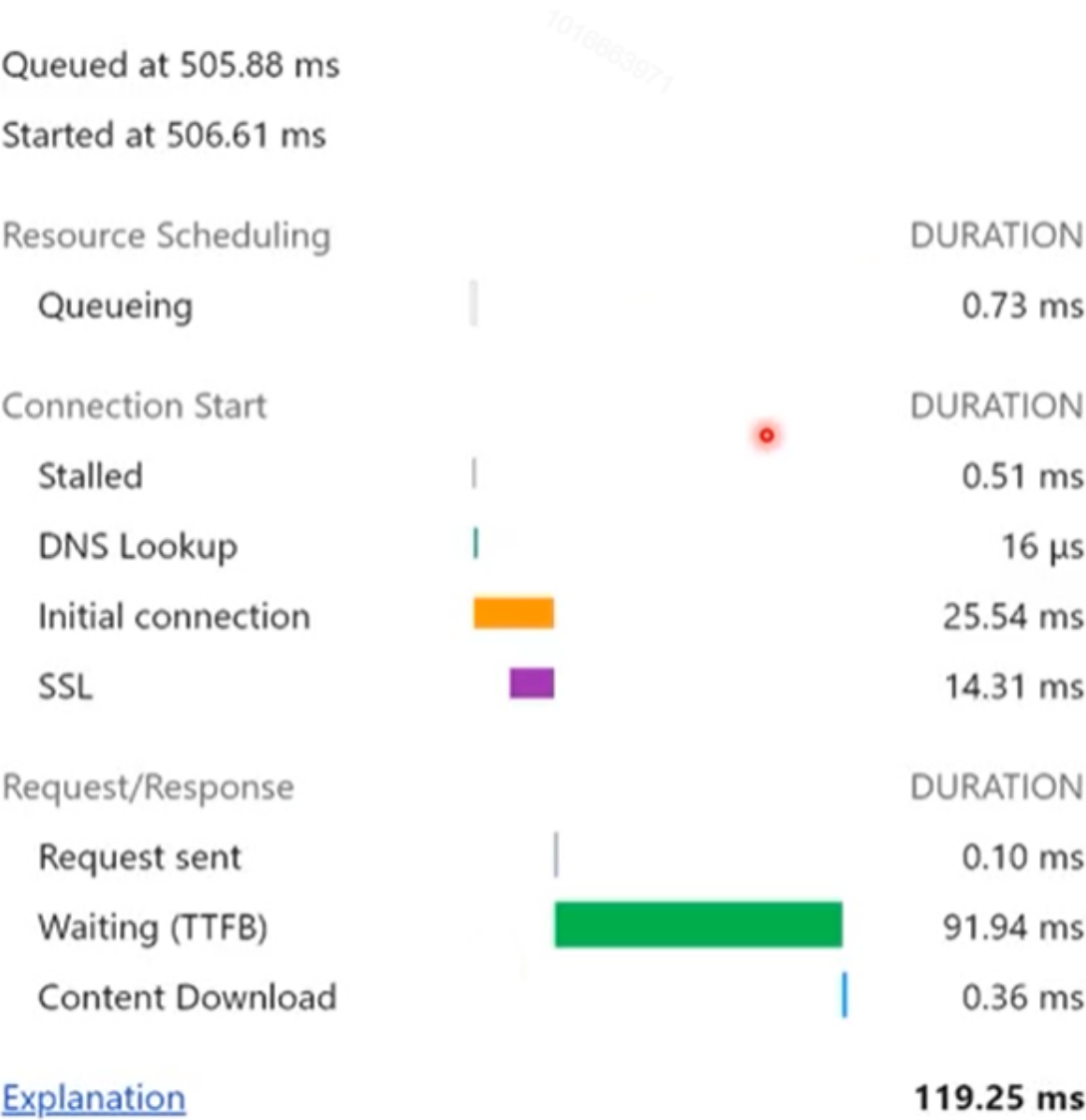


课程目标



网络指标

一个请求得消耗时间



Resource Scheduling(资源调度，浏览器通常在一个域名下只能发6个请求)

- Queueing 排队时间

Connection Start(建立连接)

- Stalled 程序准备时间
- DNS Loop DNS查询时间
- Initial connection TCP/IP三次握手的时间
- SSL 三次握手的时候, 建立SSL和TLS传输层时间

Request/Response

- Request sent 发送请求时间
- Waiting(TTFB)等待第一帧时间
- Content Download 内容下载时间

常见移动端网络性能指标

- FCP (首次内容绘制(Frist Content Page)的时间)
- TTI (用户可交互的时间)
- 跳出率 (用户进入页面, 什么操作没有, 就离开页面---可能对内容不感兴趣 或者数据加载太慢)
- CPU (CPU 消耗 输入延迟 是否卡顿)

网络优化策略

缓存

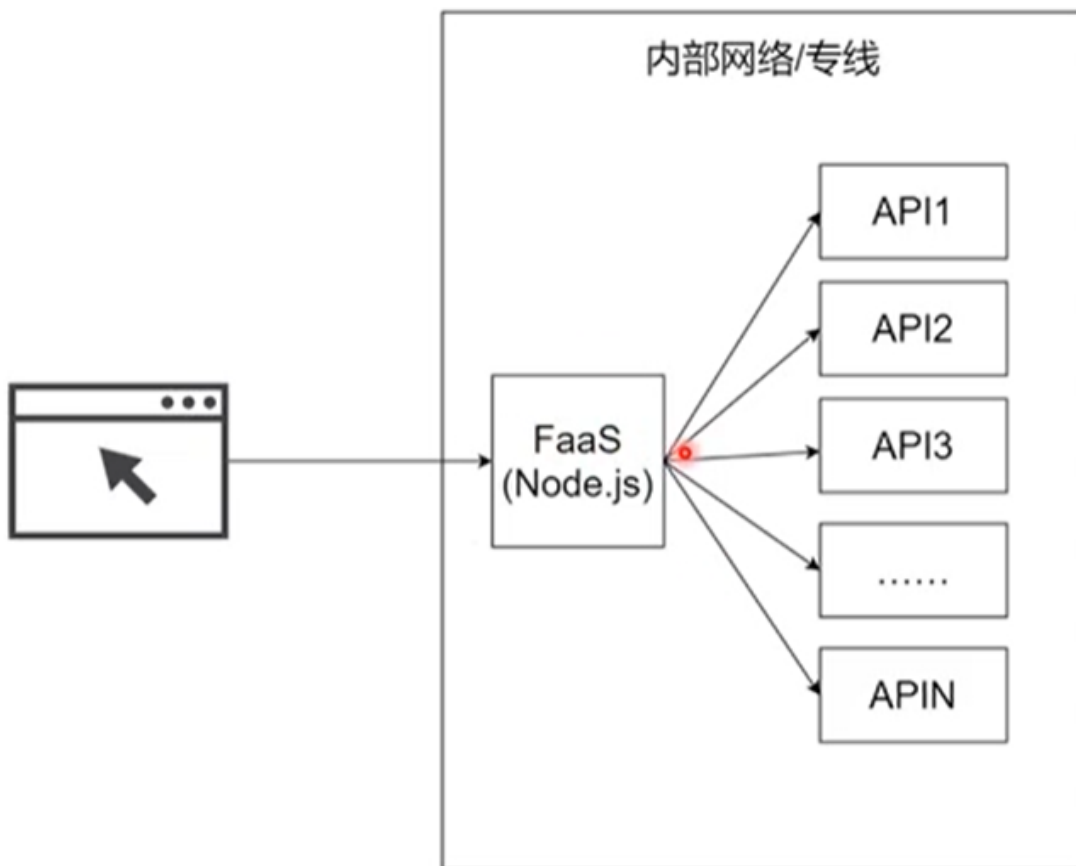
- http缓存
- 移动端缓存(native)
- 服务端缓存(redis SSR)

压缩

- js压缩 css压缩 html压缩(uglifyjs)
- Gzip(express nginx caddy)
- 合并(雪碧图--早期)

请求合并

- 一般是在服务端 在服务端部署一个Faas函数, 这个函数会和相关api都部署在一个内网, 统一有Faas函数合并请求, 多路复用, 这样就减少了单个请求的开销,
- 也可以在客户端进行请求合并



离线包

在打开APP期间将资源推送到APP的缓存中，比如你打开一个淘宝的客户端，本来这时间你去聊微信去了，淘宝的客户端是在一个后台运行的状态，这个时候淘宝的客户端可以帮你下载很多资源的，举例，你现在打开的是淘宝的首页，但是淘宝的首页需要用react.js的，你需要聊微信，淘宝的后台就帮你下载react.js，然后存到APP缓存里面(如果用户使用wifi的话，应该还好，如果是流量的话，消耗用户的过多流量，是不道德的)

预加载

- APP预加载完整容器
- H5应用预加载下一个页面的数据