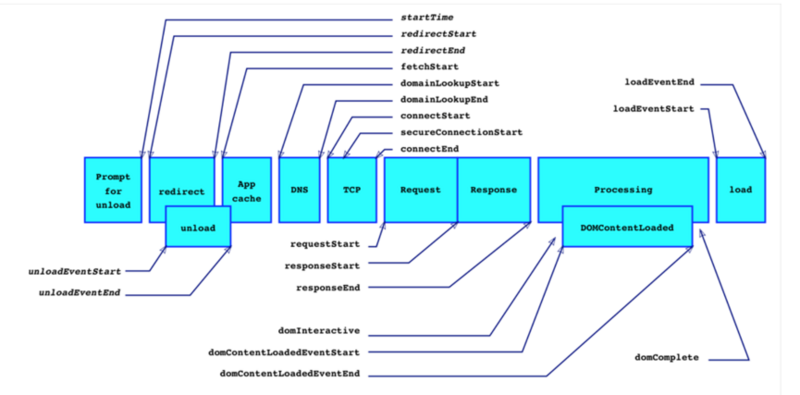
# 前端优化

页面的测速指标可以大致概括为： 白屏时间，首屏时间，可交互时间。

window.performance是w3c提供的用来测量网页和Web应用程序的性能api。其中performance timing提供了延时相关的性能信息，可以高精度测量网站性能。timing的整体结构如下图所示:



白屏时间=页面开始展示的时间点(PerformanceTiming.domLoading)-开始请求时间点(PerformanceTiming.navigationStart)

首屏时间=首屏内容渲染结束时间点(视业务具体情况而定)-开始请求时间点(PerformanceTiming.navigationStart)

可交互时间=用户可以正常进行事件输入时间点（PerformanceTIming.domInteractive）-开始请求时间点(PerformanceTiming.navigationStart)

优化结果 = 加载速度提升 + 渲染性能提升

首次内容绘制(屏时间) = 浏览器将第一个 DOM(任何文本/图像/画布/SVG等) 渲染到屏幕的时间 = window.performance.getEntriesByType('paint')

在代码中查看 paint 性能：

const observer = new PerformanceObserver((list) => {

for (const entry of list.getEntries()) {

// `entry` is a PerformanceEntry instance.

console.log(entry.entryType);

console.log(entry.startTime);

console.log(entry.duration);

}

});

// register observer for long task notifications

observer.observe({entryTypes: ["paint"]});

本地开发时性能的测量

[Lighthouse](https://developers.google.com/web/tools/lighthouse/?spm=a2c4e.11153940.blogcont598162.19.f3415522i91RgI) 和 [Web Page Test](https://www.webpagetest.org/?spm=a2c4e.11153940.blogcont598162.20.f3415522i91RgI) 为我们本地开发提供了非常好的性能测试工具，而且对于我们前面提到的各项测量标准都有较好的支持。但是，这些工具不能在用户的机器上运行，所以它们不能反映用户真实的用户体验。

用户设备中性能的测量

### **load 事件与 DOMContentLoaded 事件**

单纯地用 load 事件或者 DOMContentLoaded 事件来衡量页面性能，并不能很好地反馈出站在用户角度的页面性能。

当初始的 HTML 文档被完全加载和解析完成之后，DOMContentLoaded 事件被触发，而无需等待样式表、图像和子框架的完成加载。

当页面资源及其依赖资源已完成加载时，将触发load事件。当 onload 事件触发时，页面上所有的DOM，样式表，脚本，图片都已经加载完成了。

前端优化

在前端优化时, 特别是前端性能监控时总需要一些标准来评判, 而不总用速度快,时间短来简要概述.

有时评测指标是有一些前提的,比如 用户数量, 系统开销 在一

主要性能指标:白屏时间、首屏时间、资源加载完成时间、网页加载完成时间

比如一些指标:

1. 响应时间: 从用户角度, 2s流畅, 5s 可用, 10s 较慢
2. 吞吐量: 硬盘IO, 网格IO, CPU IO, 服务器处理能力, 客户端打开页面数量
3. 事务处理能力:

关注点:

1. 响应时间的快慢, 服务器的处理速度
2. 服务端使用情况
3. 数据库使用情况
4. 最大用户访问数量
5. 同时处理最大业务数量
6. 考察系统能否支撑7\*24时小时运转
7. 内存资源/线程资源的能否正常回收
8. 代码,算法, sql 语句设计是否合理
9. 整个系统的稳定性, 可恢复性

测试工具:

1. 基于