Web前端性能优化

1. 前端性能分析工具
2. 前端性能优化目的
3. 从用户角度而言，优化能够让页面加载得更快、对用户的操作响应得更及时，能够给用户提供更为友好的体验。
4. 从服务商角度而言，优化能够减少页面请求数、或者减小请求所占带宽，能够节省可观的资源。

总之，恰当的优化不仅能够改善站点的用户体验并且能够节省相当的资源利用。

1. 性能优化方法

前端优化的途径有很多，按粒度大致可以分为两类，第一类是页面级别的优化，例如 HTTP请求数、脚本的无阻塞加载、内联脚本的位置优化等 ;第二类则是代码级别的优化，例如 Javascript中的DOM 操作优化、CSS选择符优化、图片优化以及 HTML结构优化等等。另外，本着提高投入产出比的目的，后文提到的各种优化策略大致按照投入产出比从大到小的顺序排列。

1. HTML部分
   1. 减少HTTP请求
      1. 把多个JS请求合并为一个JS请求，把多个CSS请求合并为一个CSS请求
      2. 配置ETag

在http请求中加入ETag标记，这个标记可以在请求时检验服务器端资源是否与客户端资源相同。如果相同，只返回带304响应码的响应头，不用重新从服务器端发送页面数据，而直接使用本地缓存。

* + 1. 添加Expires头

响应头中有Expires头，设置默认过期时间30分钟。即是如果当前网页未过期，浏览器不会发请求，直接查找本地页面缓存。加速前端响应速度，减少服务器端压力。

* + 1. 压缩组件

查看http请求头参数中，有一项Accept-Encoding:gzip,deflate，响应头中有一项Content-Encoding：gzip ，这里表示请求的内容采用gzip方式进行压缩传输。减少传输文件的大小，加快页面响应

* 1. 将样式表放在顶部
  2. 使用内容分发网络(可选，主要是减轻服务器负担)
  3. 减少不必要的HTTP跳转
  4. 将脚本放在底部

浏览器的加载顺序基本是按源码从上到下加载的，把样式表放在顶部，可以加快页面样式的显示，给客户更好的体验。同理，把脚本放在底部，会避免因脚本阻塞页面内容的呈现

1. JS部分
2. 精简Javascript

对JS文件进行压缩，减少了JS的体积

1. 移除重复脚本

一个HTML文件对同一个JS文件或者CSS文件，只引入一次

1. 代码方面优化
2. 减少对DOM的操作
3. 减少循环遍历
4. 慎用with，避免用eval和Function
5. 合理封装方法，较少代码量
6. CSS部分
7. 利用CSS3和图标字体
8. 利用CSS Sprites合并图片，减少请求数的一个好办法
9. Inline Images

使用 data: URL scheme的方式将图片嵌入到页面或 CSS中，如果不考虑资源管理上的问题的话，不失为一个好办法。如果是嵌入页面的话换来的是增大了页面的体积，而且无法利用浏览器缓存。使用在 CSS中的图片则更为理想一些。

1. Lazy Load Images(图片懒加载)

这条策略实际上并不一定能减少 HTTP请求数，但是却能在某些条件下或者页面刚加载时减少 HTTP请求数。对于图片而言，在页面刚加载的时候可以只加载第一屏，当用户继续往后滚屏的时候才加载后续的图片。这样一来，假如用户只对第一屏的内容感兴趣时，那剩余的图片请求就都节省了。 有啊首页 曾经的做法是在加载的时候把第一屏之后的图片地址缓存在 Textarea标签中，待用户往下滚屏的时候才 “惰性” 加载。