

02

OPEN ORIENTED

凹凸实验室

主流电商网站前端性能浅析

范立之

自我介绍



范立之（荔枝）



京东消费

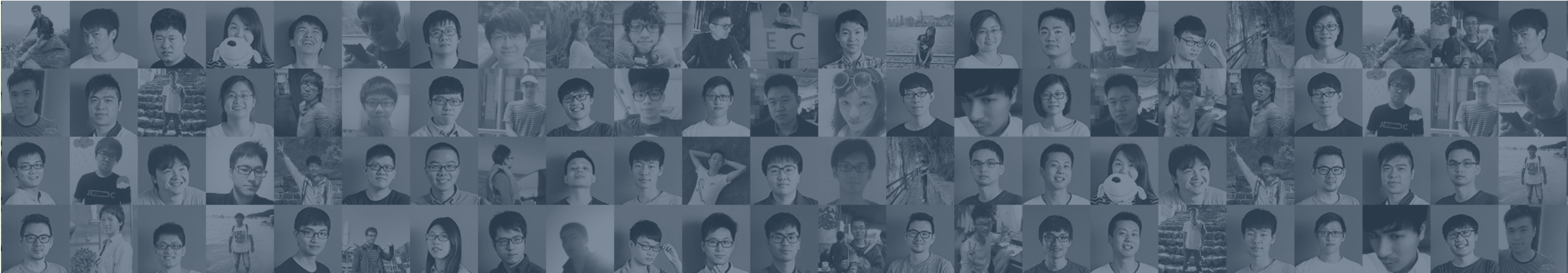
京东生鲜



4 年



<https://aotu.io/>





















redirections

重定向(Redirect)就是通过各种方法将各种网络请求重新定个方向转到其它位置（如：网页重定向、域名的重定向、路由选择的变化也是对数据报文经由路径的一种重定向）

	example.com		
	example.com	m.example.com	
	example.com	www.example.com	m.example.com

Redirections

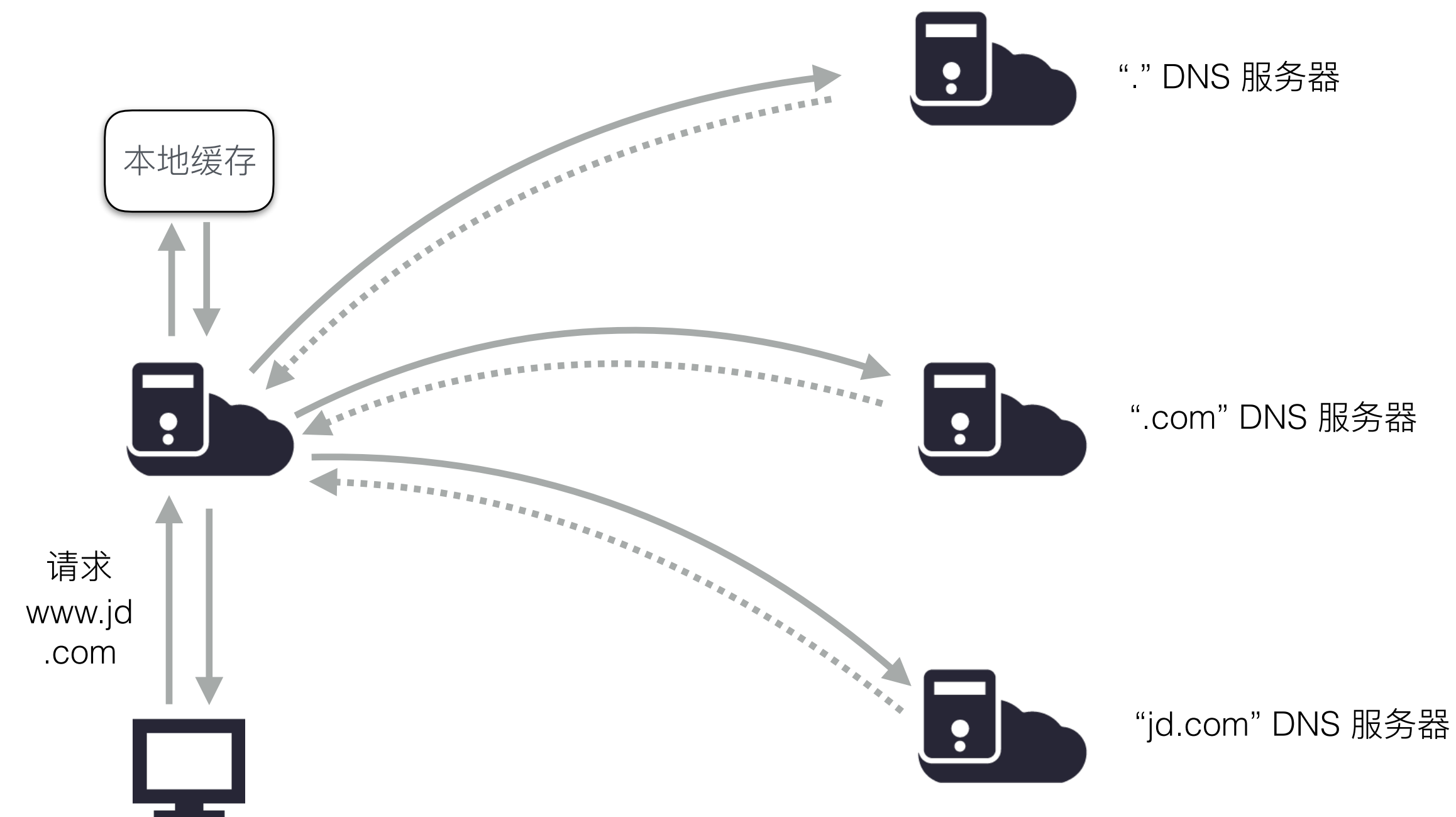
业界使用情况
* 移动设备访问域名

	http://amazon.cn	http://www.amazon.cn		
	http://tmall.com	http://www.tmall.com	https://www.tmall.com	
	http://jd.com	http://www.jd.com	https://www.jd.com	
	http://taobao.com	http://www.taobao.com	https://www.taobao.com	
	http://m.taobao.com	https://m.taobao.com		



DNS

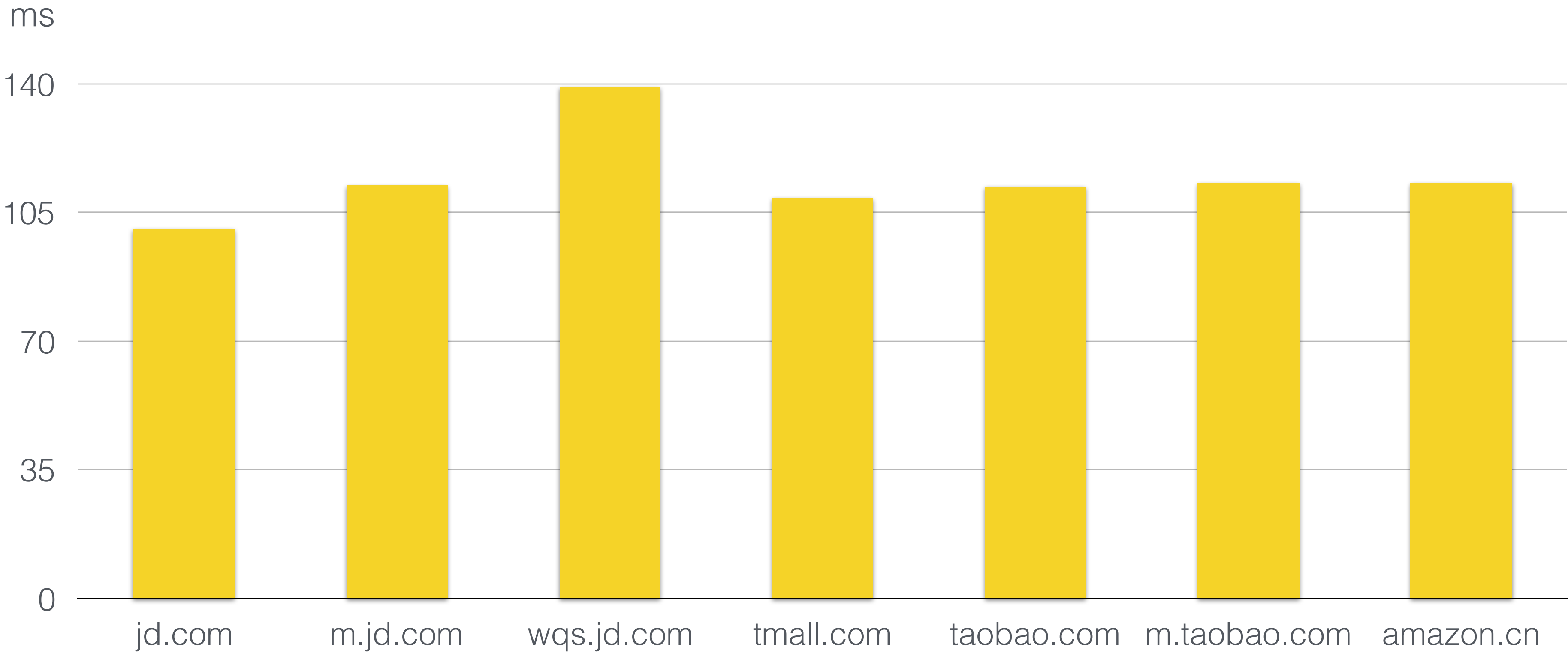
域名系统（英文：Domain Name System，缩写：DNS）是因特网的一项服务。它作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。DNS使用TCP和UDP端口53。当前，对于每一级域名长度的限制是63个字符，域名总长度则不能超过253个字符。



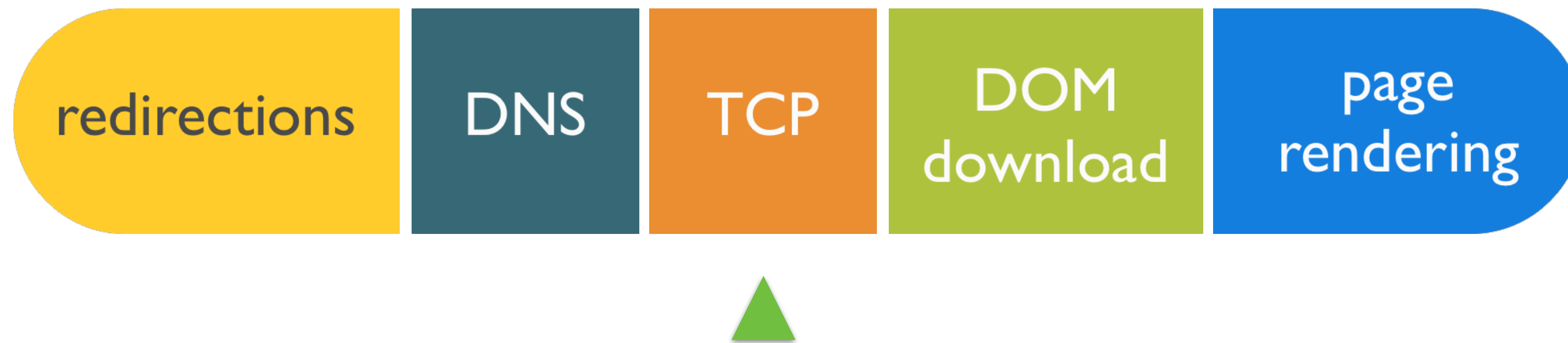
`www.jd.com` -> 浏览器 -> hosts -> Local DNS（服务商） -> 根域名服务器（全球13台，1+12） -> 返回主域名服务器

- 1: 好的域名解析商
- 2: CDN厂商的域名解析服务
- 3: 拆分域名, 多个域名, 不管是在IE6和IE8的情况下, 拆分域名, 都会使网站的打开速度变快, 50M以下 4-5, 100-500M 8-10, ie6-7 同一个域名只能同时并行2个资源
- 4: 域名预获取 `<link rel="dns-prefetch" href="//s1.mi.com">` dns-prefetch IE9 +, `<meta http-equiv='x-dns-prefetch-control' content='off'>` 避免解析, 还有一个值是on

DNS 解析时间，单位毫秒 😊

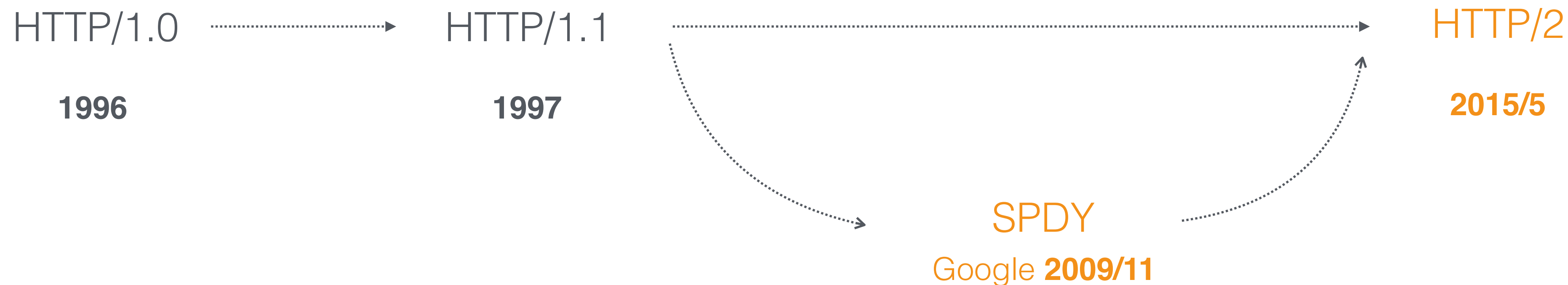


* 测试方式奇云测，ce.cloud.360.cn



TCP

传输控制协议（英语：Transmission Control Protocol，缩写为 TCP）是一种面向连接的、可靠的、基于字节流的传输层通信协议，HTTP为TCP的应用层



*SPDY当前并不是一个标准协议，但SPDY的开发组已经开始推动SPDY成为正式标准（现为互联网草案），HTTP/2主要以SPDY技术为主。

注：Chrome 51 不支持 NPN 和 SPDY，Ngnix 也支持HTTP/2废除了SPDY。

http1.1



3.21s

http2



1.43s

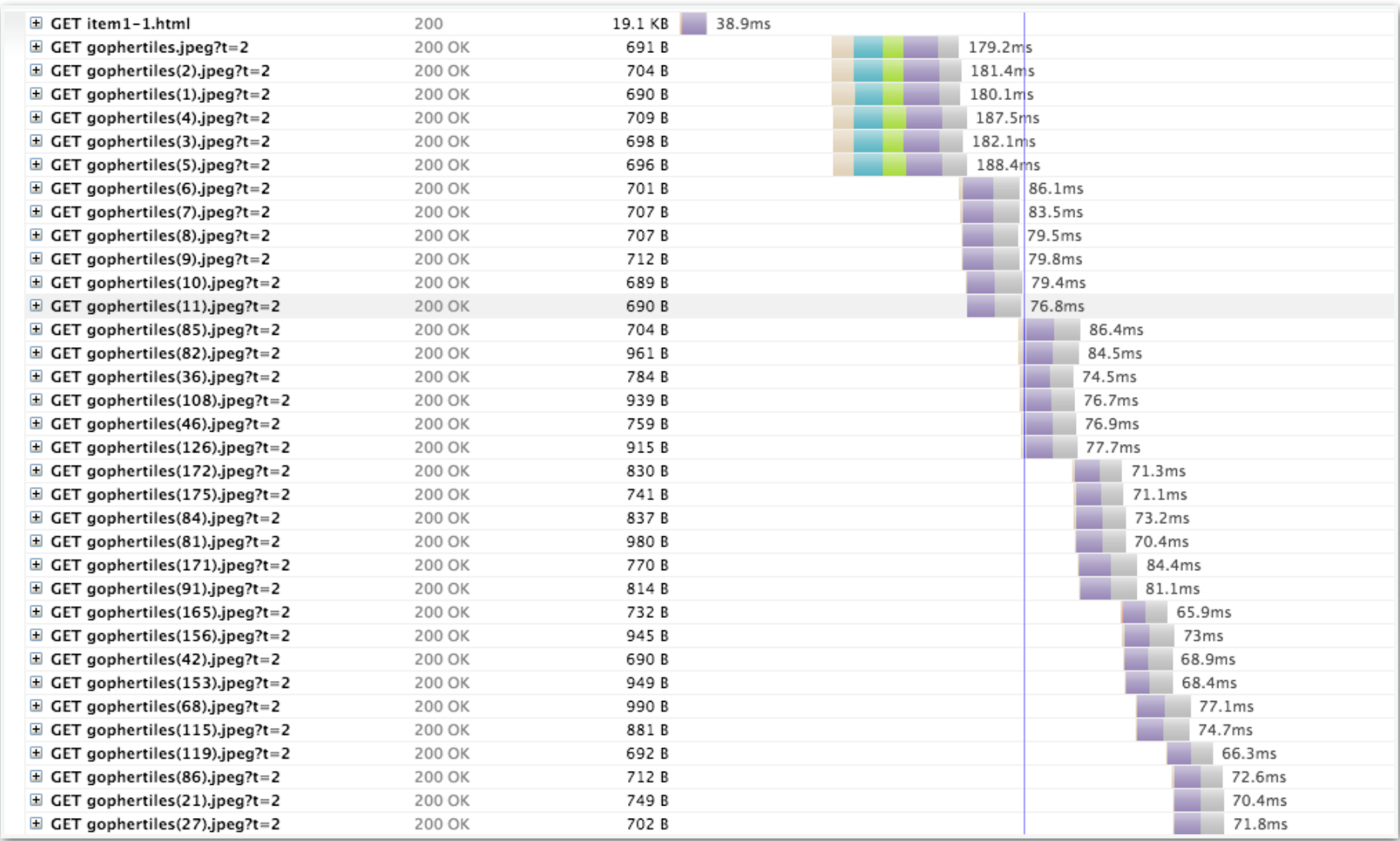
The screenshot shows the Chrome DevTools Network tab. At the top, there's a search bar with 'gophertiles(5)' and a filter dropdown set to 'All'. Below the search bar is a waterfall chart showing the timing of network requests. The first request, 'gophertiles(5)', is highlighted. Below the chart is a table of network requests:

Name	Status	Protocol	Type	Initiator	Size	Time
gophertiles(5).jpeg	200	h2	jpeg	item1.html:211	733 B	73 ms
gophertiles(5).jpeg?t=2	200	http/1.1	jpeg	item1.html:28	918 B	129 ms

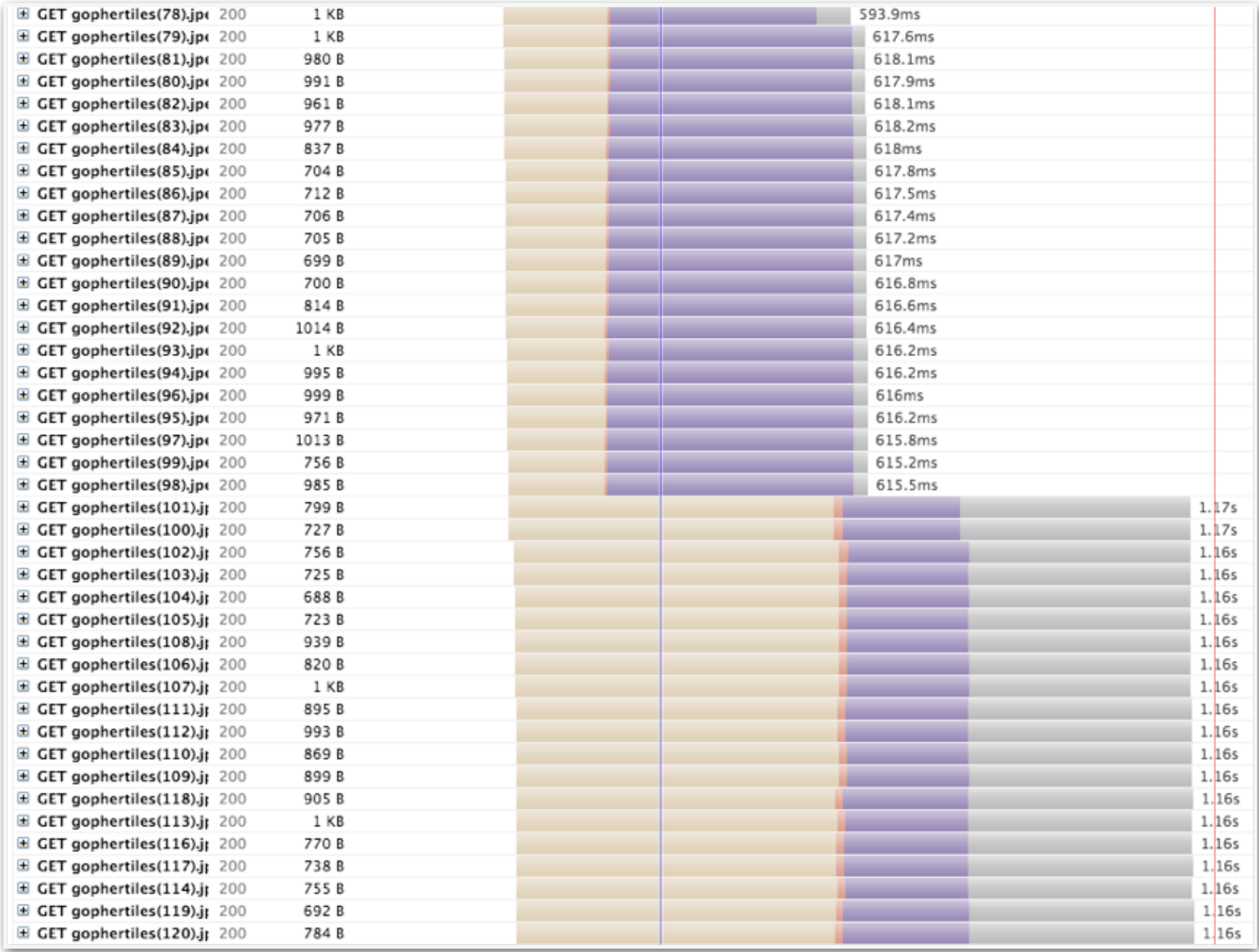
A tooltip is visible over the second request, showing the URL: `http://l-zhi.com:8080/img/gophertiles(5).jpeg?t=2`.

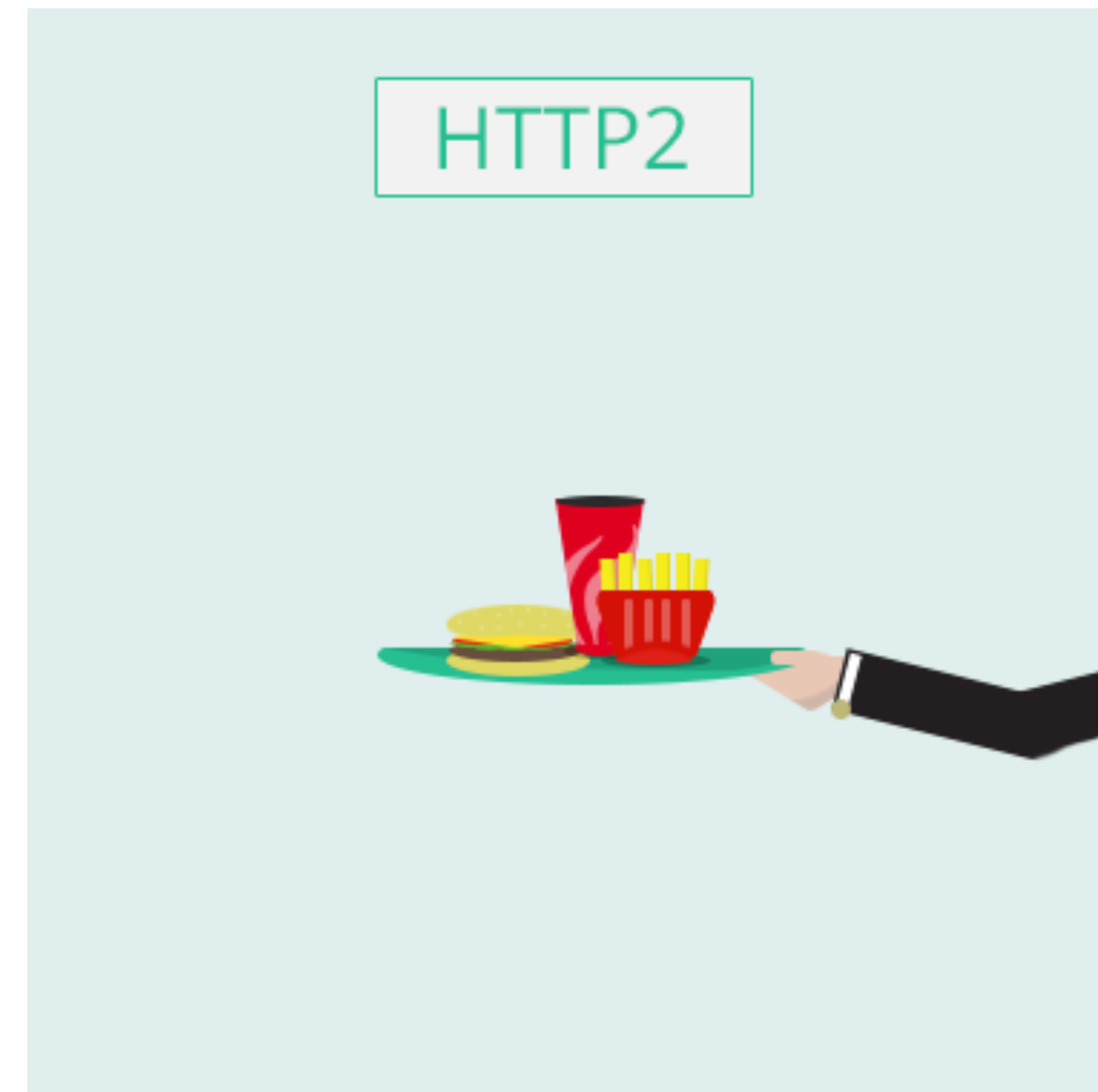
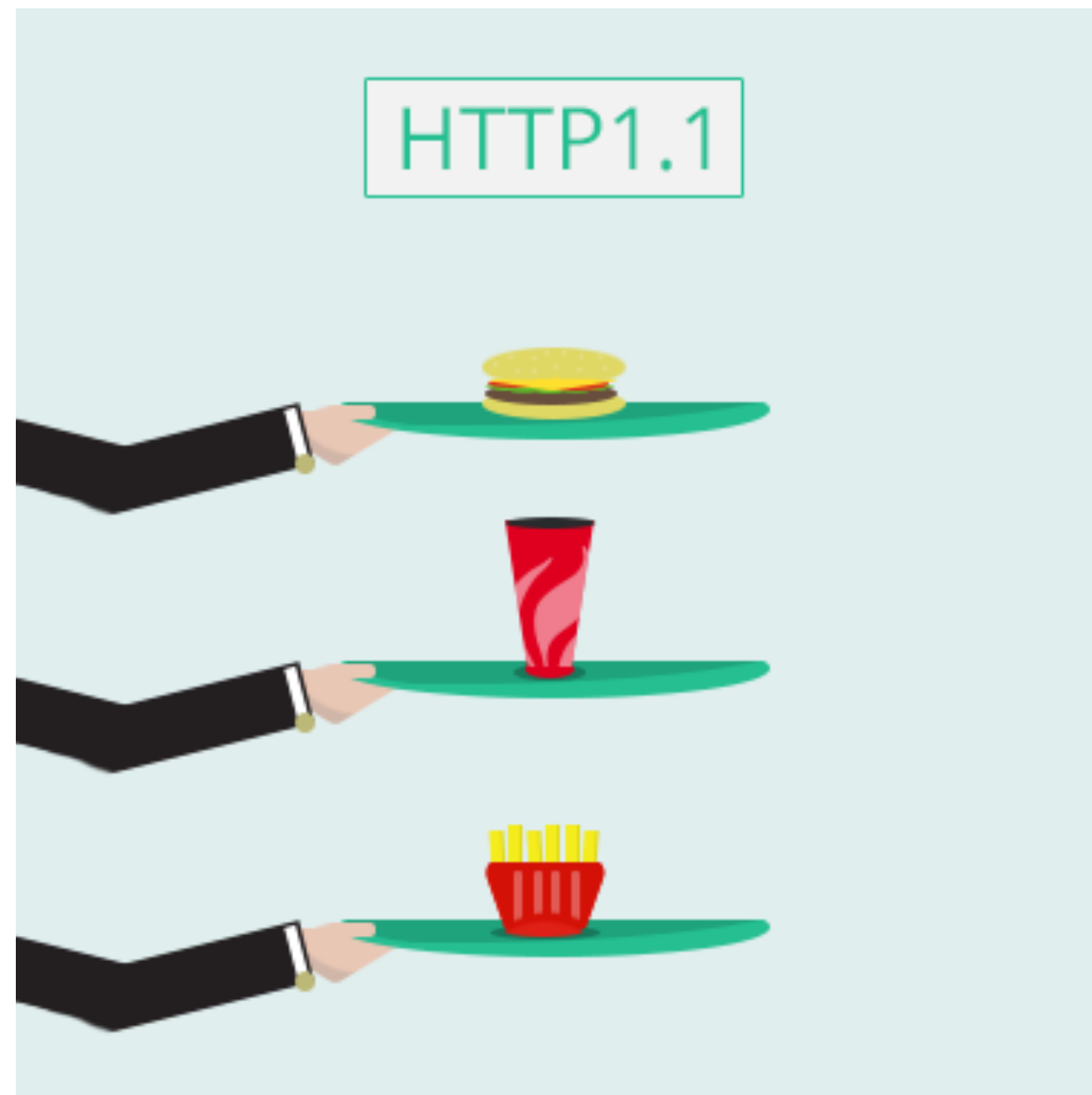
每张大图由169张小图组成，上面截取其中一张的对比请求

HTTP/1.1



HTTP/2

















HTTP/2 ServerPush

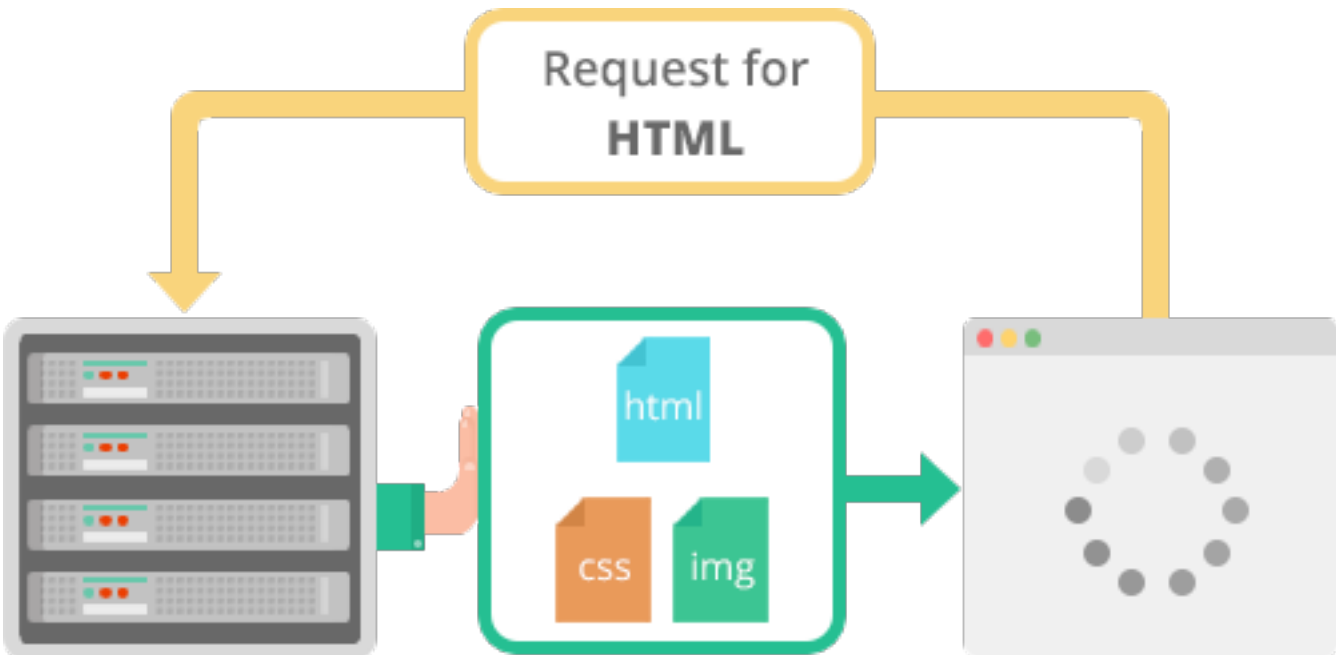
server push

传统的资源请求

 item2.html	200 OK	docu...	Other	896 B 660 B	153 ms 115 ms		
 zepto.js /js	200 OK	script	item2.html... Parser	16.5 KB 58.2 KB	118 ms 66 ms		
 underscore.js /js	200 OK	script	item2.html... Parser	14.8 KB 51.8 KB	150 ms 101 ms		
 backbone.js /js	200 OK	script	item2.html... Parser	19.8 KB 70.1 KB	153 ms 100 ms		









server push

 serverpush	200	docu...	Other	745 B 580 B	227 ms 226 ms		
 underscore.js	200	script	serverpush... Parser	14.5 KB 51.8 KB	10 ms 6 ms		
 backbone.js	200	script	serverpush... Parser	19.5 KB 70.1 KB	10 ms 6 ms		
 zepto.js	200	script	serverpush... Parser	16.3 KB 58.2 KB	9 ms 6 ms		



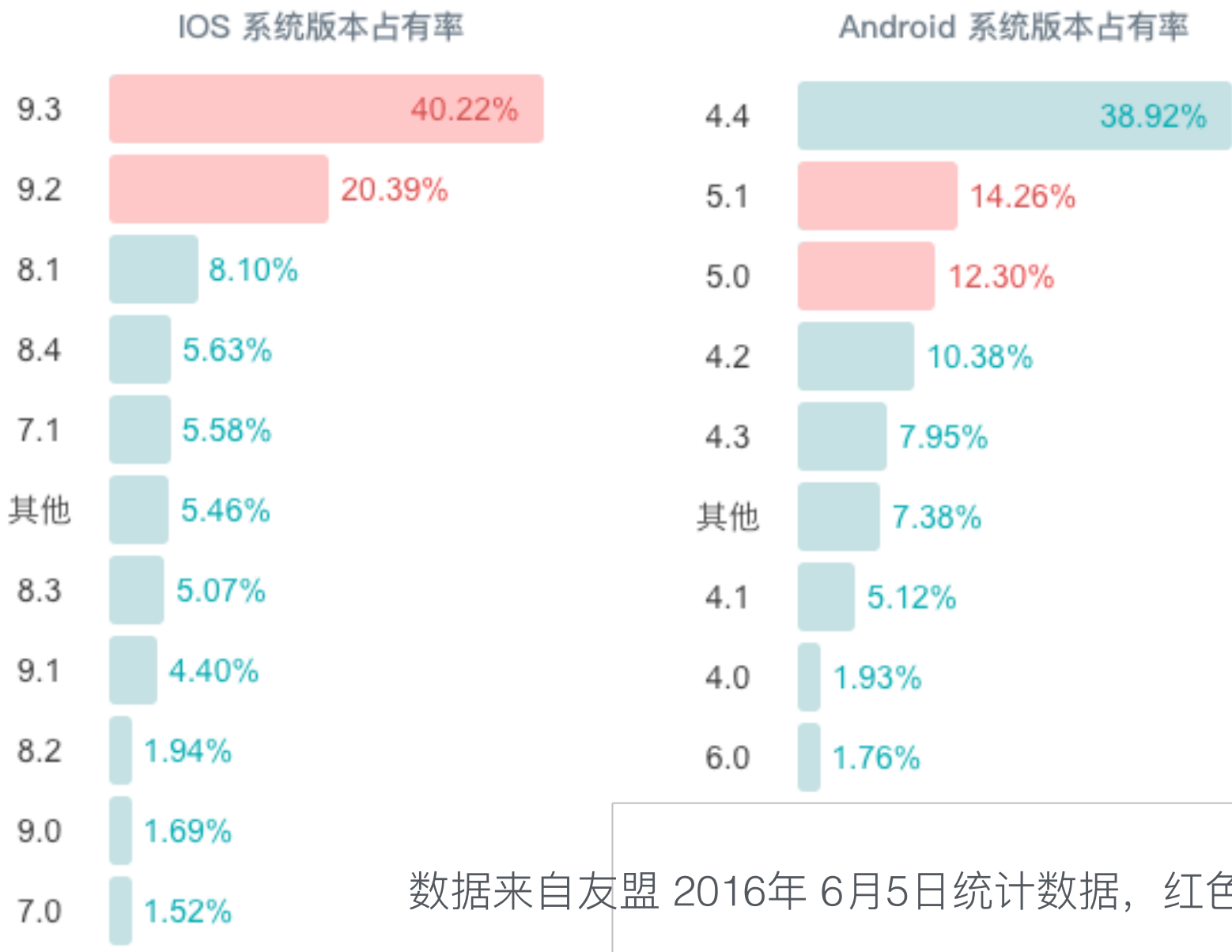
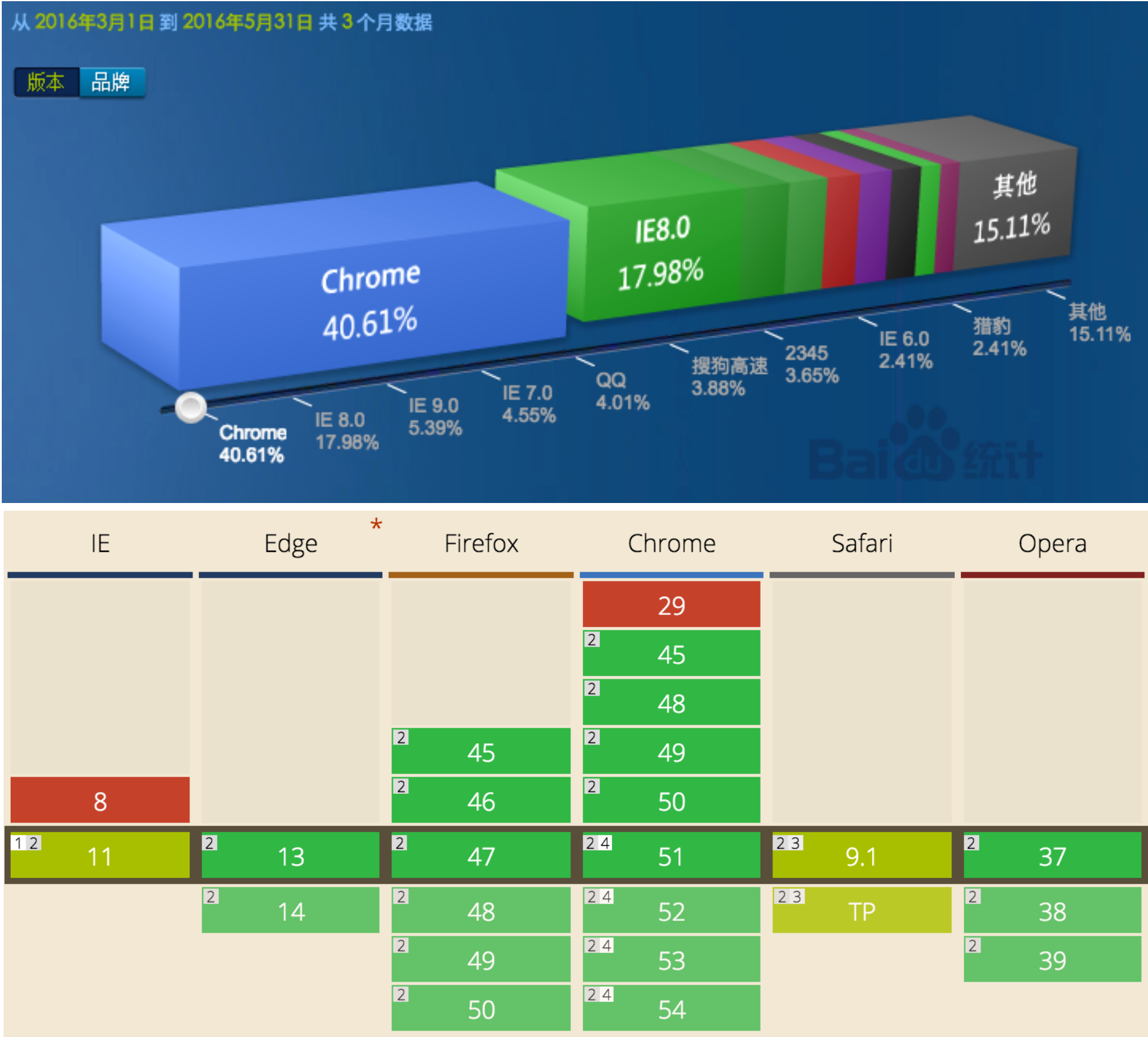
HTTP/2

业界使用情况

	HTTPS & HTTP1.1	
	HTTPS & HTTP1.1	
	HTTPS & HTTP1.1 & SPDY3.1 & HTTP2	
	HTTPS & HTTP1.1 & SPDY3.1 & HTTP2	

天猫淘宝，在Chrome 51 不支持 SPDY 和 HTTP/2，需要升级 OpenSSL 1.0.2













移动端 26%以上 PC端 41%以上



数据来自友盟 2016年 6月5日统计数据，红色为支持版本



权重： 30%

	redirections	DNS	TCP	
				2
				1
				3
				2

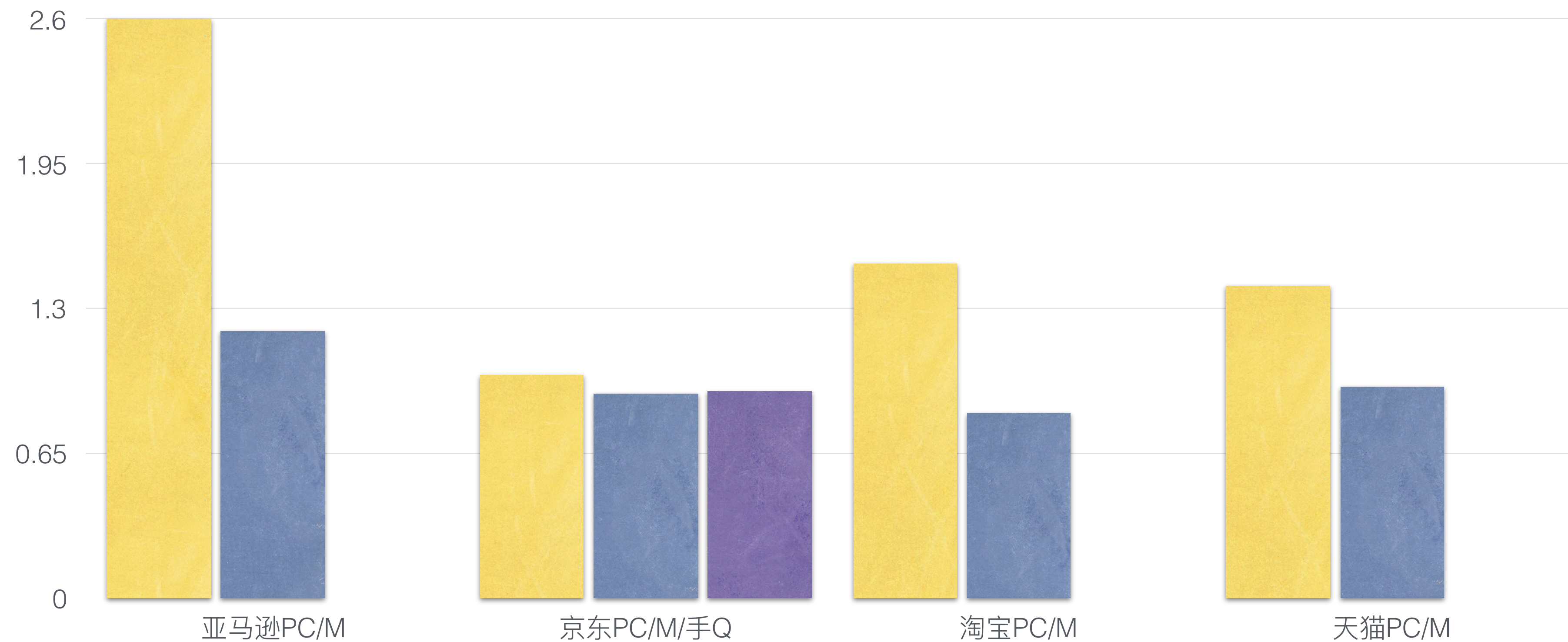


DOM download

DOM
download

首屏资源加载，包括HTML、CSS、JavaScript、Image、资源缓存等

DOMContentLoaded (单位秒)



*网络条件: DSL(2 Mb/s)

DOM download

业界首屏加载情况

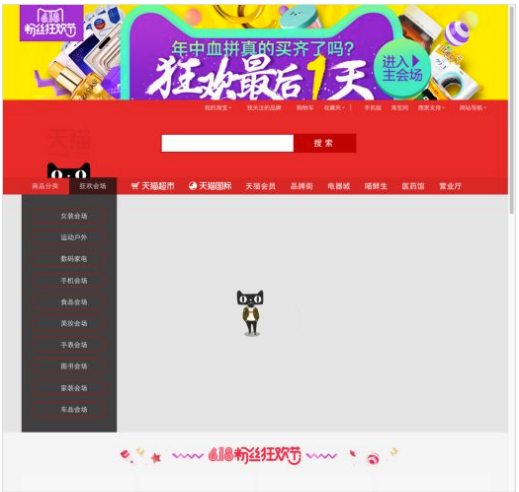


859ms



2.65s

	?ie=UTF8...	200 OK	http/1.1	docum...	Other	124 KB 693 KB	864 ms 85 ms	
--	-------------	-----------	----------	----------	-------	------------------	-----------------	--



823ms



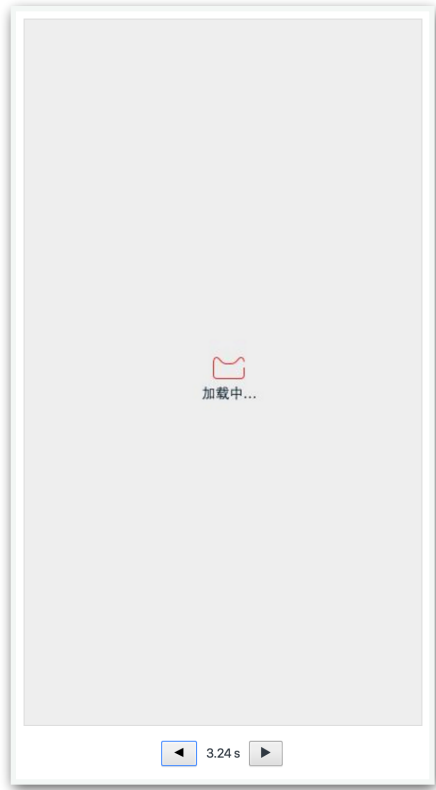
2.70s

	www.tmal...	200 OK	http/1.1	docum...	Other	50.0 KB 230 KB	20 ms 12 ms	
--	-------------	-----------	----------	----------	-------	-------------------	----------------	--

*网络条件：DSL(2 Mb/s)

DOM download

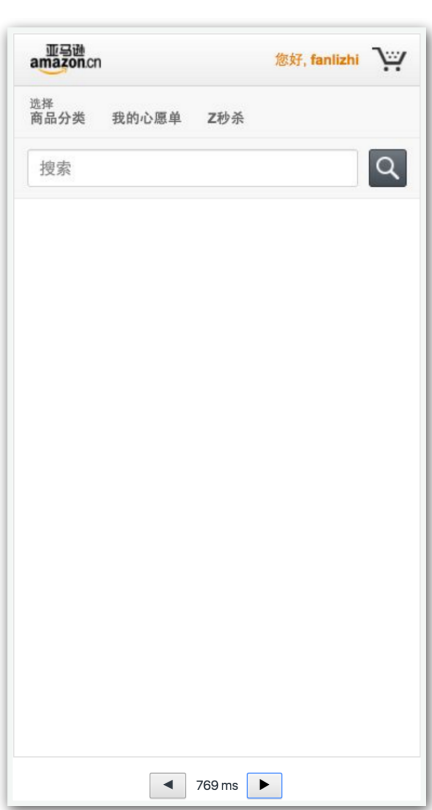
业界首屏加载情况



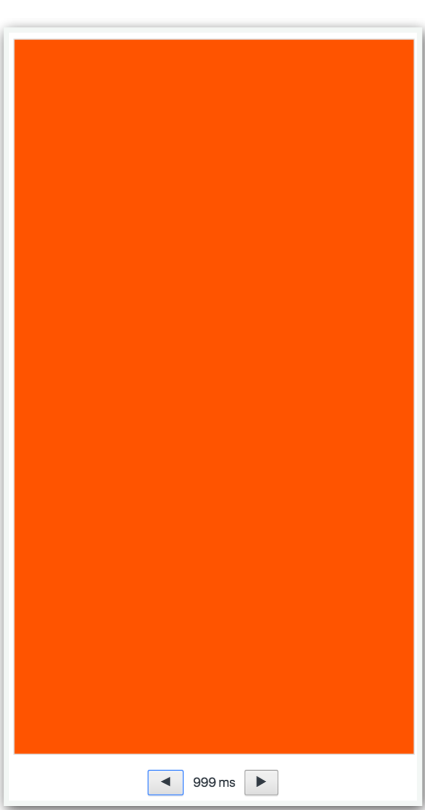
1.20s



1.30s



0.769s



1s

*网络条件：DSL(2 Mb/s)

DOM download

	总资源数	JS	CSS			总资源数	JS	CSS	
	228	16	9			95	15	6	
	128	18	2			106	6	2	
	144	46	6			85	12	4	
	200	55	6			35	5	0	

DOM download






02



```
<div data-ctrl-name="scrollview" data-dpr="1" data-ctrl-id="1461125296280-1" data-direction="vertical">
  <div class="top-fixed" style="left: 0px; top: 0px; width: 100%; display: block;">...</div>
  <div class="scroll-wrap">
    <div class="scroll-content" style="transform: translate3d(0px, 0px, 0px); height: 3299.61px;">
      <div id="n7098cd" class="n2dsk" style="display: block; position: relative; z-index: 0;">...</div>
      <div id="content">
        <section data-group="1">
          <div data-template-name="tbanner" style="position: relative;">
            <div style="box-sizing: border-box; line-height: 0; width: 10rem; height: 3.125rem; " class="UIView">...</div>
          </div>
          <div data-template-name="tentrance" style="position: relative;">...</div>
          <div data-template-name="toutiao2" style="position: relative;">...</div>
          <div data-template-name="gray_border_home" style="position: relative;">...</div>
          <div data-template-name="rushbuy42" style="position: relative;">...</div>
          <div data-template-name="tcheap3" style="position: relative;">...</div>
        </section>
      </div>
    </div>
  </div>
```

整个页面没有css外链















减少资源大小，节省加载时间，图片一致都是页面流量的大头，所以优化图片显得尤为重要。WebP最初在2010年发布，目标是减少文件大小，但达到和JPEG格式相同的图片质量，希望能够减少图片档在网络上的发送时间。

	PNG 原图	PNG 无损	WebP 无损	PNG 有损(75%)	WebP 有损(75%)
精细度					
Alpha 透明	支持 (8-bit)				
图片体积	33.8 kb	30.1 kb	24.1 kb	9.34 kb	7.16 kb
体积减少	--	10.9%	28.7%	72.4%	78.9%
备注			备选方案		推荐方案

WebP 比 JPG 小 20% ~ 30%，有损压缩将更小

DOM download - 图片优化 WebP

业界使用状况

	PC	Mobile
	 	 
		
		
		

京东图床已经全面支持webp，在等待业务逻辑上的修改

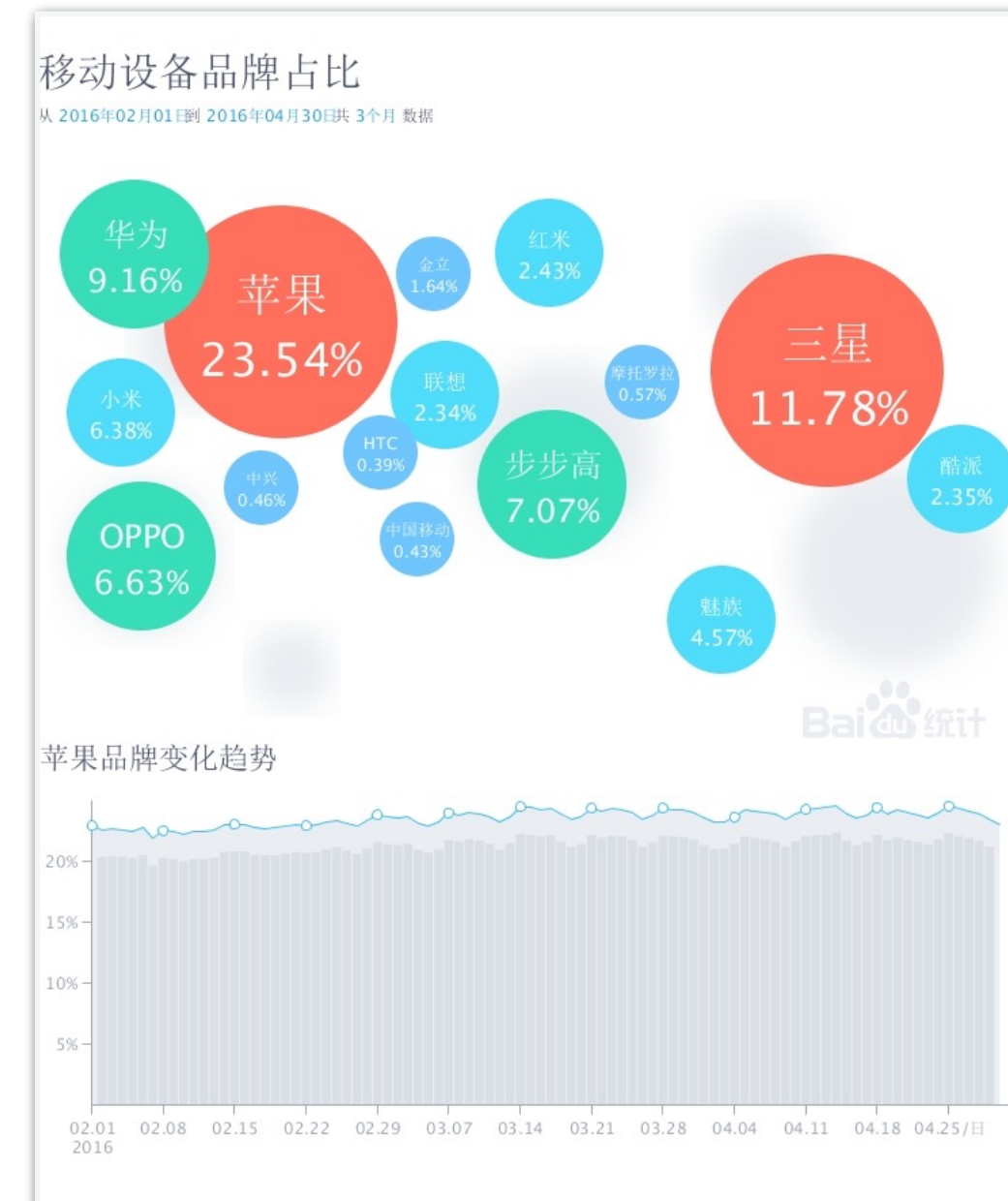
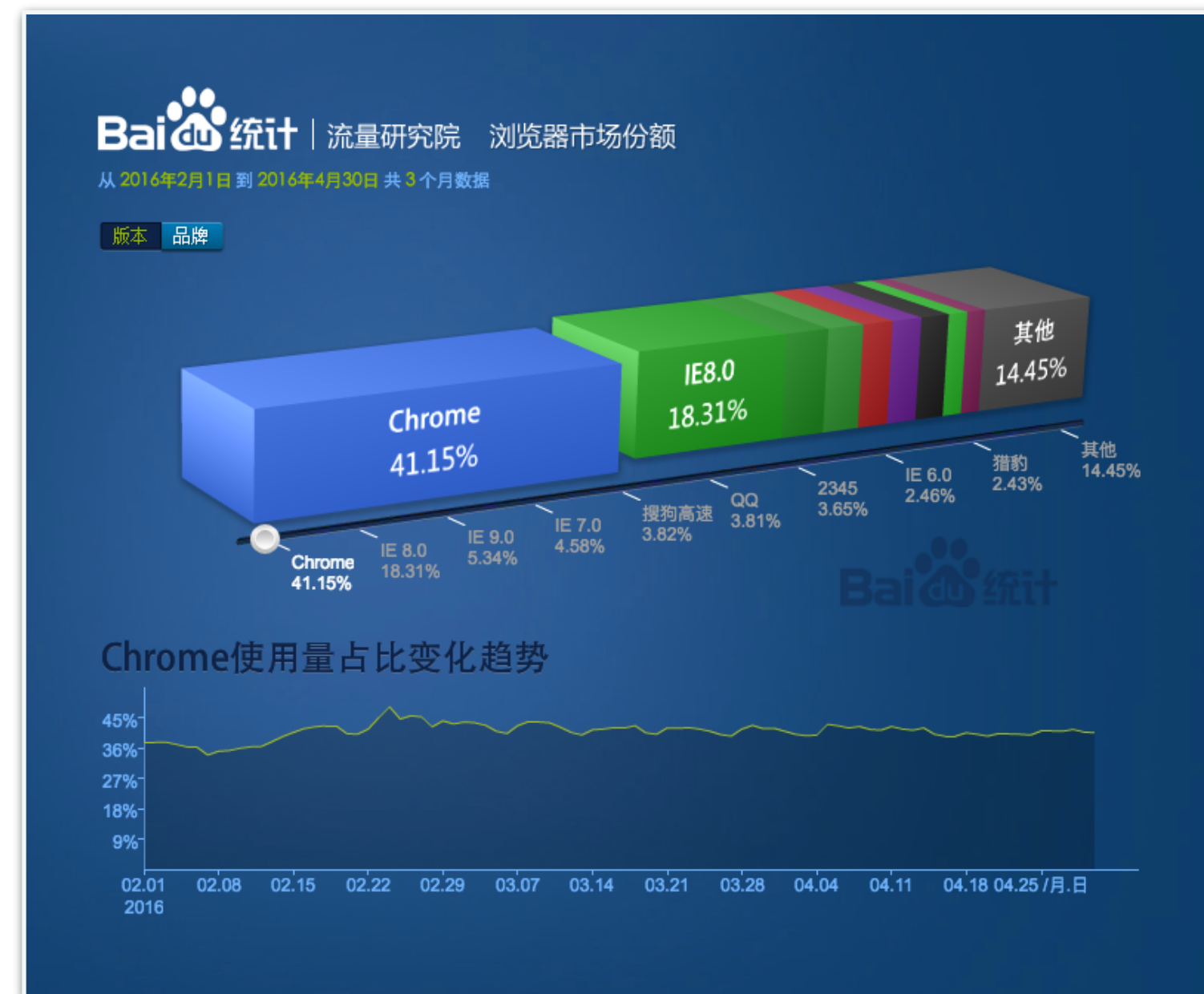
DOM download - 图片优化 WebP

兼容性

02

Chrome + Android支持 WebP，可通过代码配置优雅降级，不支持WebP用回JPG。

移动端 **70%以上** PC端 **41%以上**



百度统计2016PC端移动端浏览器占比

DOM download - 图片优化 WebP

解决方案，图片使用webp

WebP 的使用

检测方案：

1. 客户端通过 JS 检测

```
function check_webp_feature(feature, callback) {
    var kTestImage = {
        ossy: "UklGRiIAAABXRUJQVlA4IBYAAAwAQcdASoBAAEADsD+JaQAA3AAAAAA",
        lossless: "UklGRhoAAABXRUJQVlA4TA0AAAAvAAAAEAcQERGIiP4HAA==",
        alpha: "UklGRkoAAABXRUJQVlA4WAoAAAAQAAAAAAAAAAAAQUxQSAwAAAARBxAR/Q9ERP8DAABWUDggGAA",
        animation: "UklGRlIAAABXRUJQVlA4WAoAAAAASAAAAAAAAAAAAQU5JTQYAAAD/////AABBTk1GJgAAAAAA"
    }

    var img = new Image()
    img.onload = function() {
        var result = (img.width > 0) && (img.height > 0)
        callback(feature, result)
    }
    img.onerror = function() {
        callback(feature, false)
    }
    img.src = "data:image/webp;base64," + kTestImages[feature]
}
```

成功后，可以在 cookie 或者 localStorage 中植入能力检测字段

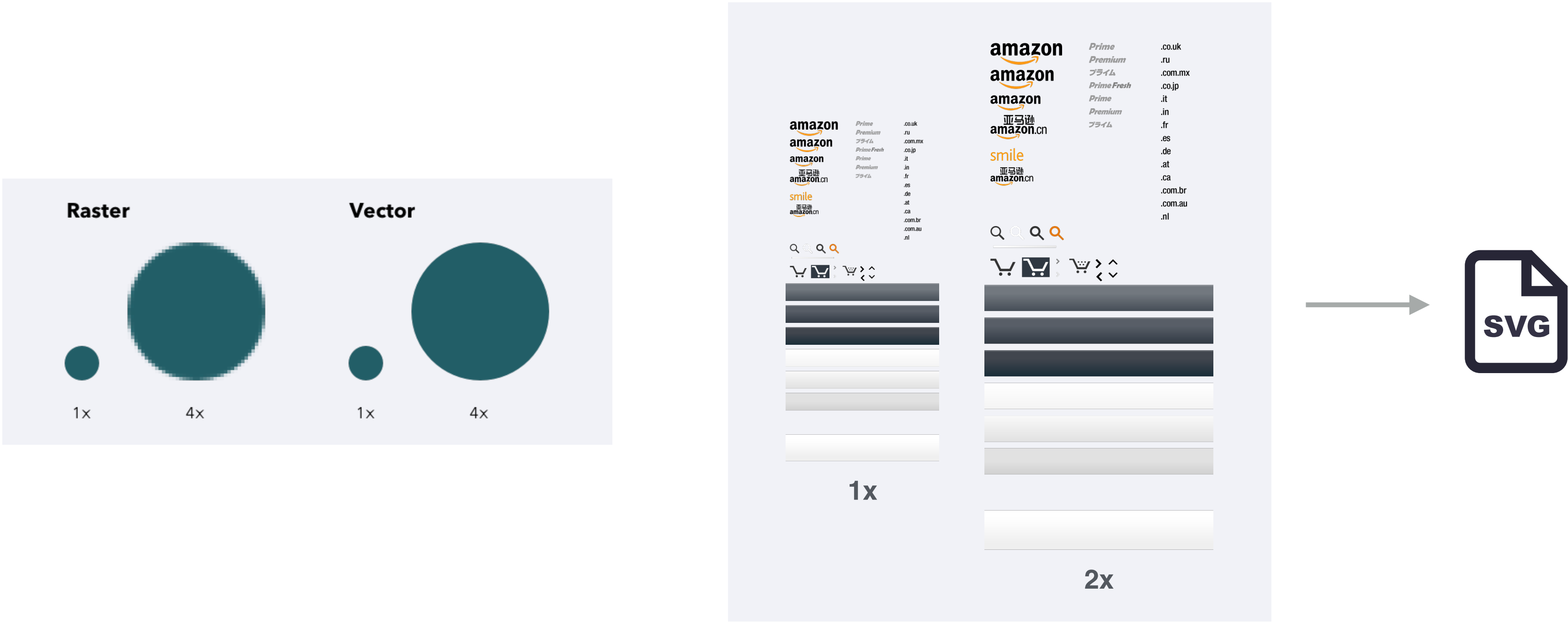
1. 用 HTML5 的 <picture> 元素

```
<picture>
  <source type="image/webp" srcset="images/butterfly.webp">
  
</picture>
```

IOS 9.3+ 支持，Android 不支持

DOM download - SVG Sprite















图片Icon需要适配不同DPI的屏幕，就要准备2~3套icon来适配，以免模糊，提升维护成本和适配的代码量，因为SVG的矢量属性，所以能很好的解决这个问题，并且还能节省资源大小。



不用分两套雪碧图，也不需要讲究排列完全语义化引用

DOM download - SVG Sprite

业界使用状况


	PC	Mobile
	 	 
		
		
		

DOM download - SVG Sprite

行内 Demo

外链 Demo

 icon-phone-hang-up

 icon-address-book

 icon-mug

 icon-spoon-knife

 icon-star-half

 icon-star-full

 icon-grin2

 icon-flickr2

 icon-flickr3

 icon-dice

DOM download - SVG Sprite

* **svg > defs > symbol > title > path**

```
<svg style="position: absolute; width: 0; height: 0;" width="0" height="0" version="1.1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
  <defs>
    <symbol id="icon-phone-hang-up" viewBox="0 0 16 16">
      <path class="path1" d="..."></path>
    </symbol>
    <symbol id="icon-address-book" viewBox="0 0 16 16">
      <path class="path1" d="..."></path>
      <path class="path2" d="..."></path>
      <path class="path3" d="..."></path>
      <path class="path4" d="..."></path>
      <path class="path5" d="..."></path>
    </symbol>
    ...
  </defs>
</svg>
```

行内

```
<body>
...
<svg class="icon icon-mug"><use xlink:href="#icon-phone-hang-up"></use></svg>
</body>
```

外链

```
<body>
<svg class="icon icon-mug"><use xlink:href="sprite.svg#icon-address-book"></use></svg>
</body>
```

./sprite.svg

DOM download - SVG Sprite

兼容性

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			29						
			45					14.3	
8			48					4.4	
29		45	49	9	36	8.4		4.4.4	
11	13	46	50	9.1	37	9.2	8	50	50
	14	47	51	TP	38	9.3			
		48	52		39				
		49	53						

* IE 9-11 do not support referencing external files via <use xlink:href>

其他

通过Ajax异步加载，使SVG外链文件，支持IE9-11的方法 <https://github.com/Keyamoon/svgxuse>

QQ浏览器X5内核支持SVG

SVG 压缩： <https://github.com/svg/svgo>

DOM download - SVG Sprite

生成SVG Sprite

02

<http://jkphl.github.io/svg-sprite/>



Node

Grunt

Gulp

Gulp

```
'use strict';

var gulp    = require('gulp'),
    svgSprite = require('gulp-svg-sprite'),
    plumber  = require('gulp-plumber'),
    baseDir  = 'svg/base/dir', // <-- Set to your SVG base directory
    svgGlob  = '**/*.svg',    // <-- Glob to match your SVG files
    outDir   = 'output/dir',   // <-- Main output directory
    config   = {};

gulp.task('svgsprite', function() {
  return gulp.src(svgGlob, {cwd: baseDir})
    .pipe(plumber())
    .pipe(svgSprite(config)).on('error', function(error){ console.log(error); })
    .pipe(gulp.dest(outDir))
});
```

其他方式: <https://icomoon.io/app/#/select>

DOM download - SVG Sprite



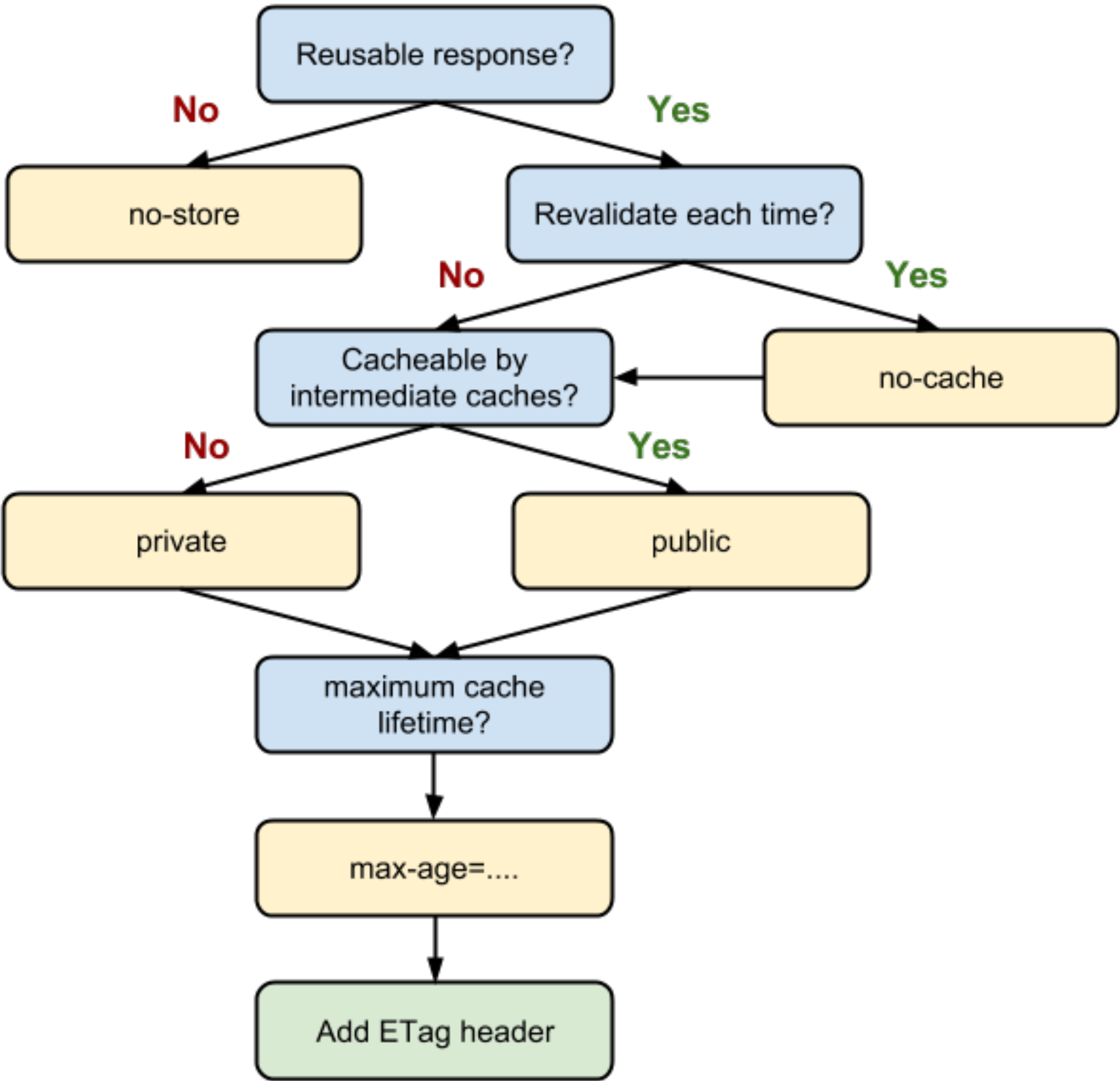
京东移动端Web产品



1. 体验更好图标不会模糊
2. 不用合成多种尺寸的雪碧图
3. 减少资源请求
4. 方便前端修改页面

DOM download - HTTP缓存

Cache-Control 策略



*左侧黄色部分为可赋值

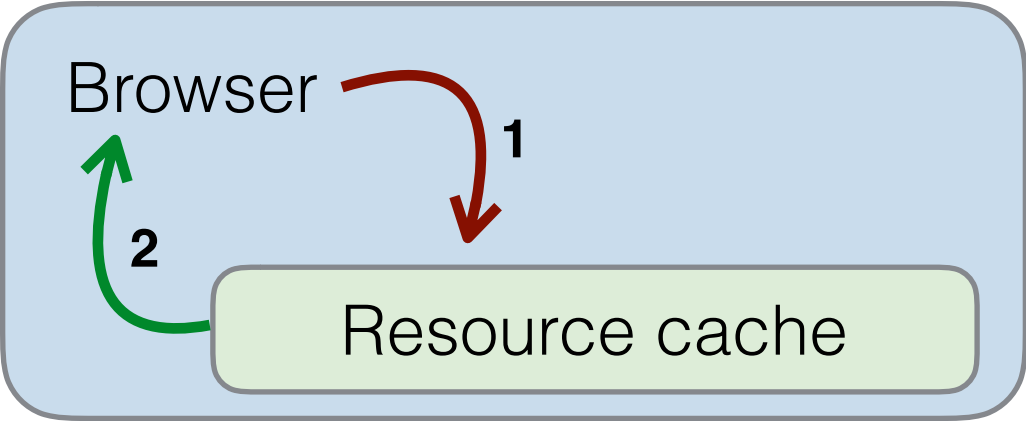
Cache-Control 指令	说明
max-age=86400	浏览器和任何中继缓存均可以将响应（如果是`public`的）缓存长达一天（60 秒 x 60 分 x 24 小时）
private, max-age=600	客户端浏览器只能将响应缓存最长 10 分钟（60 秒 x 10 分）
no-store	不允许缓存响应，每个请求必须获取完整的响应。

DOM download - HTTP缓存

实际例子

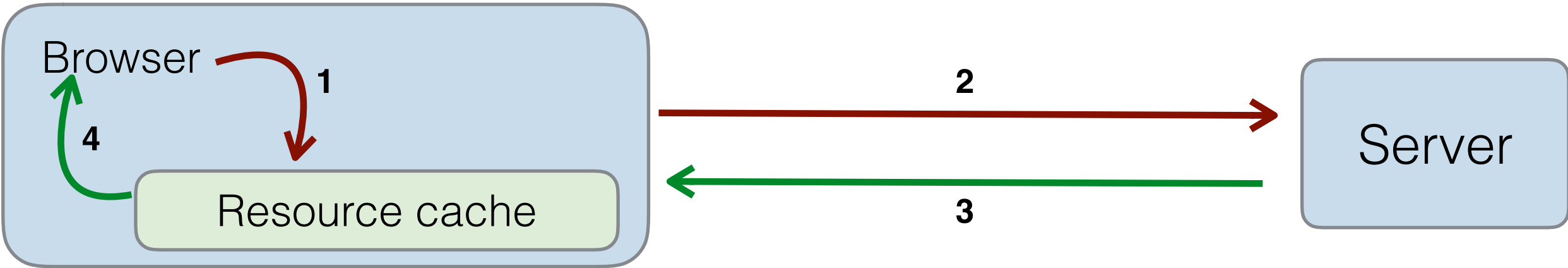
浏览器从本地缓存获取

 zepto.js /js	200 OK	http/1.1	(index):8 Parser	(from cache)	34 ms 21 ms
---	-----------	----------	---------------------	--------------	----------------

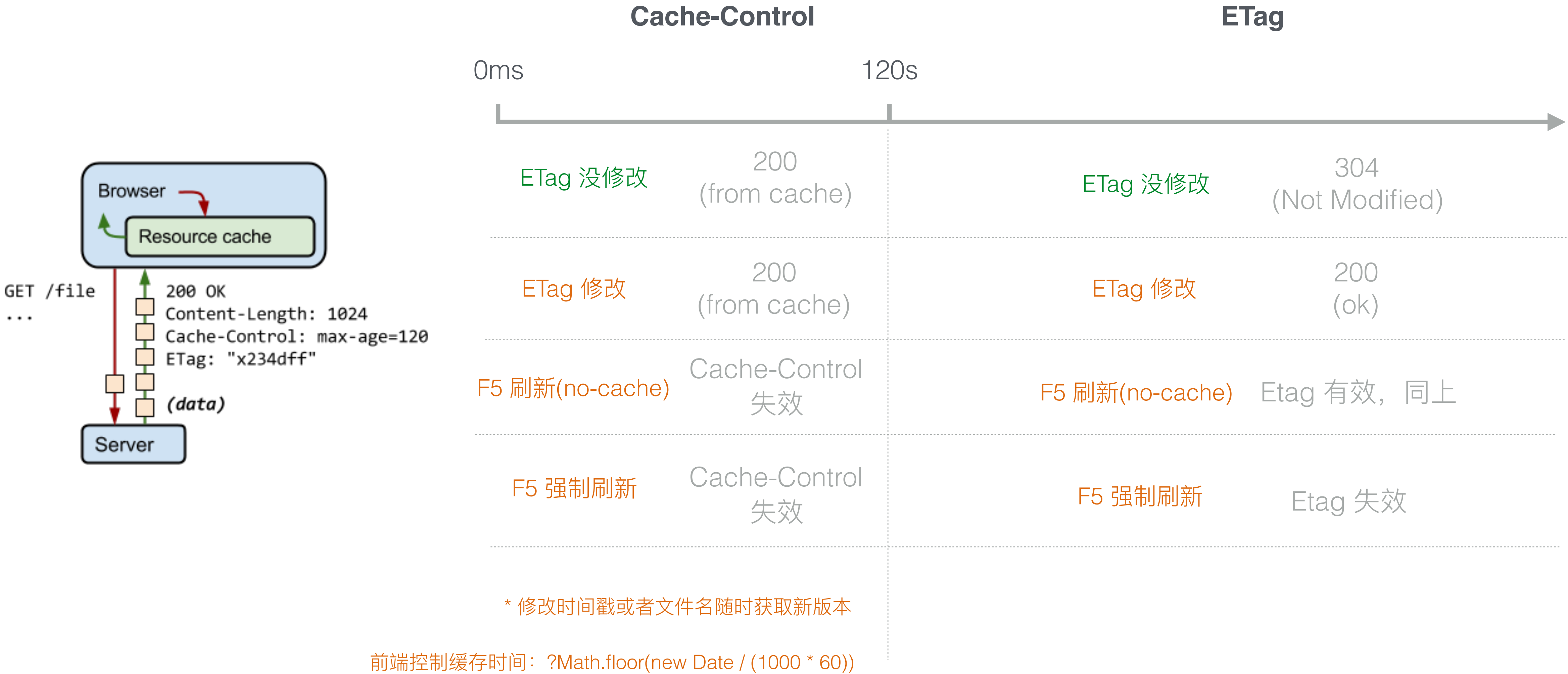


请求服务器查看资源是否修改，没有修改，走本地缓存

 zepto.js /js	304 Not Modified	http/1.1	(index):8 Parser	193 B 58.2 KB	53 ms 46 ms
---	---------------------	----------	---------------------	------------------	----------------



DOM download - HTTP缓存



DOM
download

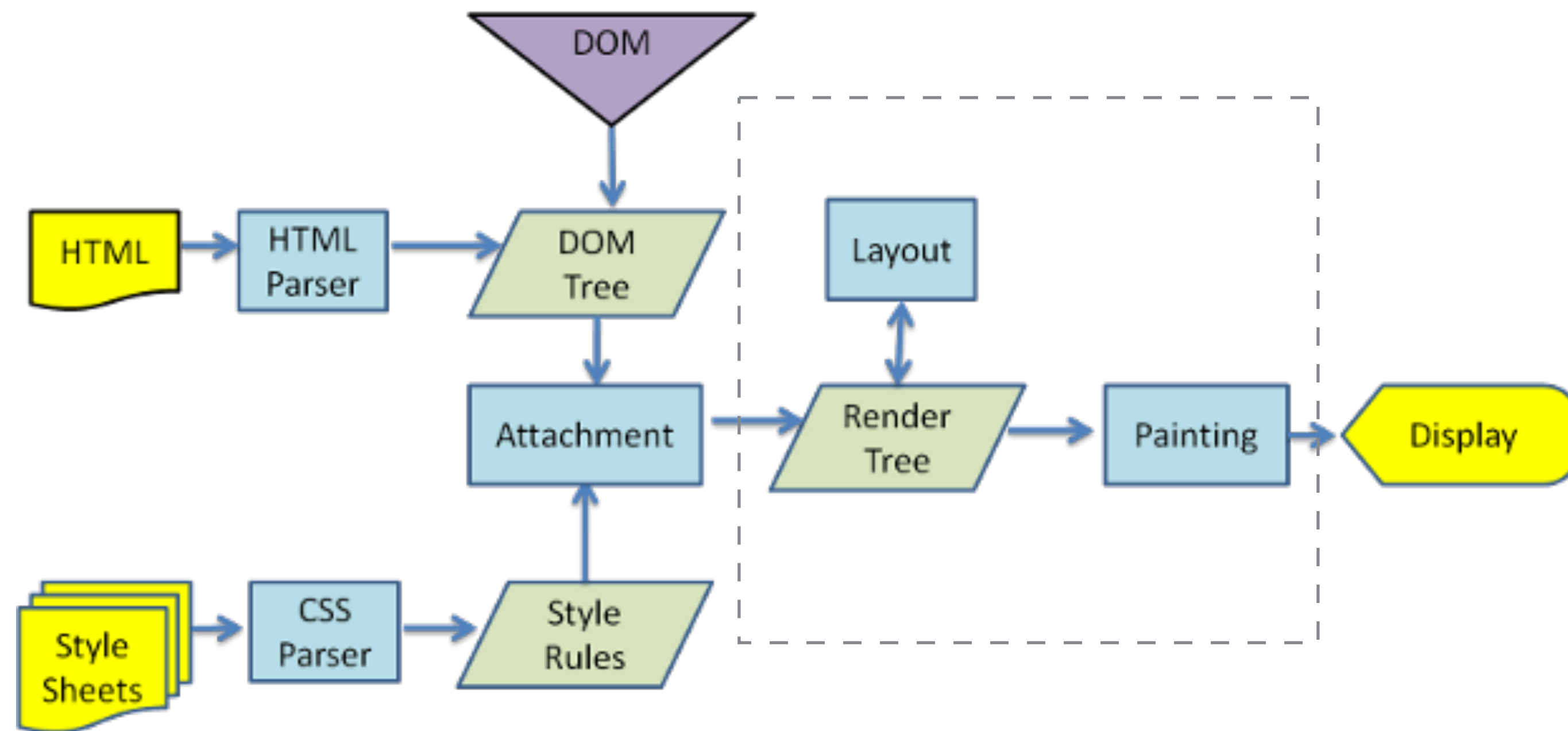
权重：40%

	Resource	HTTP缓存	WebP	SVG	
					2
					2
					4
					4



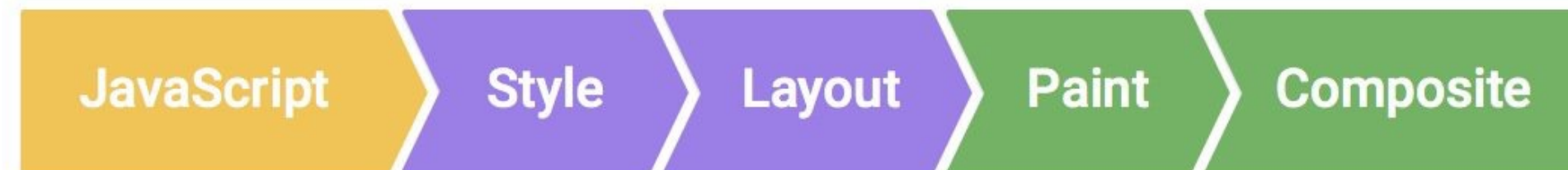
Page rendering

page
rendering



Page rendering

60fps和设备刷新率: 每秒显示帧数为60帧



也就是说, 浏览器对每一帧画面的渲染工作需要在16毫秒 ($1\text{秒} / 60 = 16.67\text{毫秒}$) 之内完成



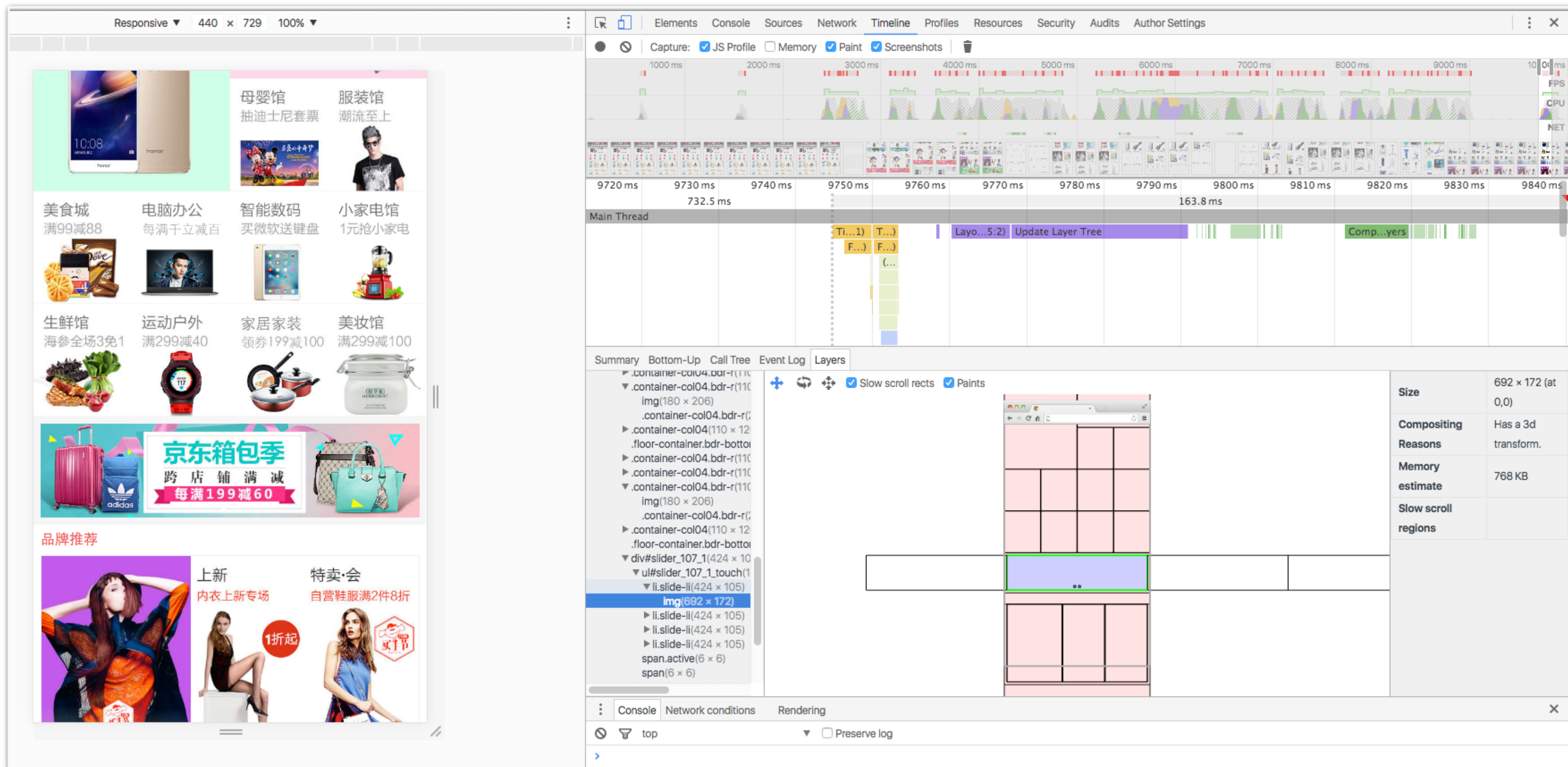
如果你修改一个DOM元素的“paint only”属性, 比如**背景图片、文字颜色或阴影等**, 这些属性不会影响页面的布局, 因此浏览器会在完成样式计算之后, 跳过布局过程, 只做绘制和渲染层合并过程。



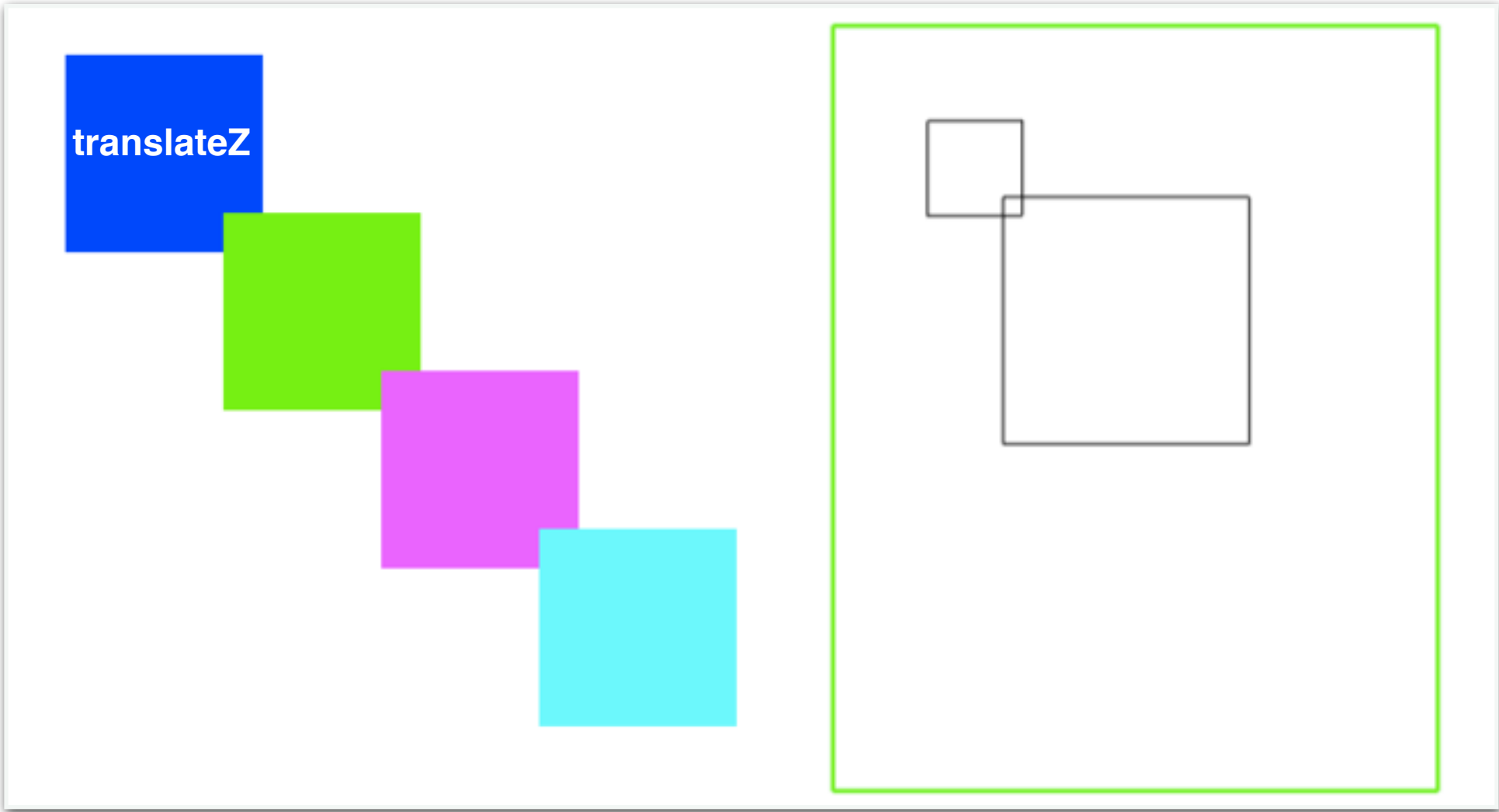
- CSS 3D属性 (perspective transform)
- 使用加速视频解码的 <video> 元素
- 拥有 3D (WebGL) 上下文或加速的 2D 上下文的 <canvas> 元素
- 混合插件(如 Flash)
- 对自己的 opacity 做 CSS 动画或使用一个动画变换的元素
- 拥有加速 CSS 过滤器的元素
- 一个元素拥有一个子元素, 该子元素在自己的层里
- 元素有一个 z-index 较低且包含一个复合层的兄弟元素(换句话说就是该元素在复合层上面渲染)
- will-change

<https://csstriggers.com/>

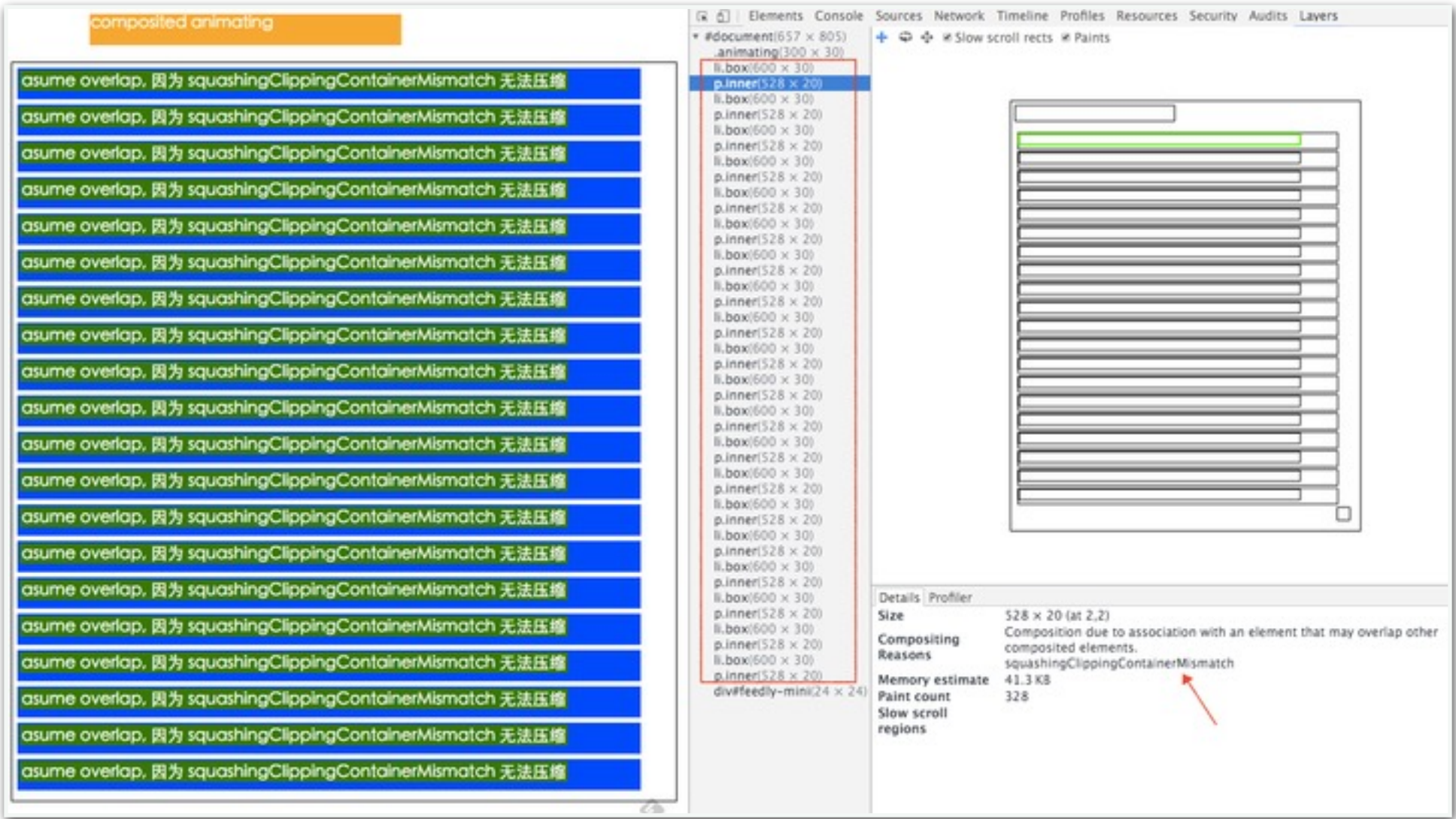
02



Page rendering



Demo



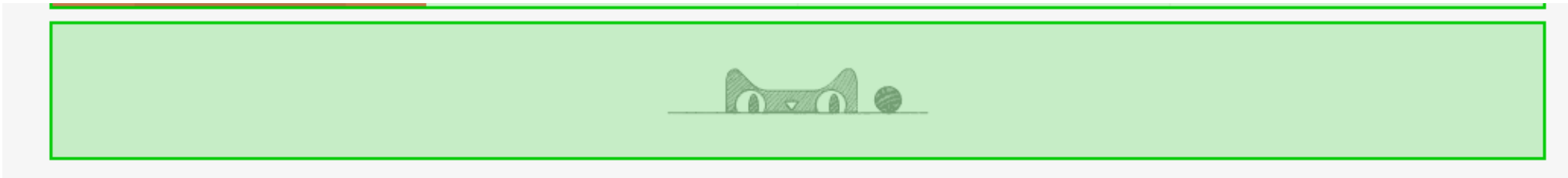
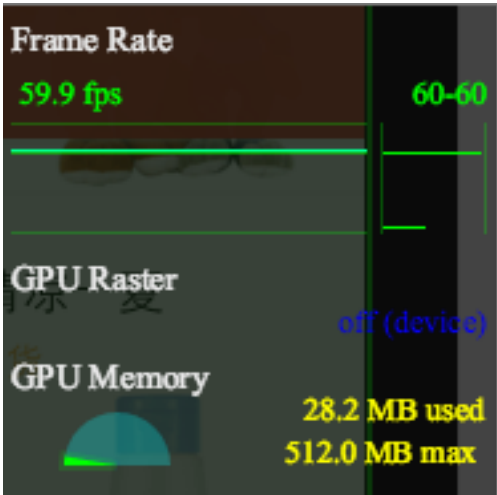
Demo



天猫移动端 快速出现loading，然后二次渲染页面



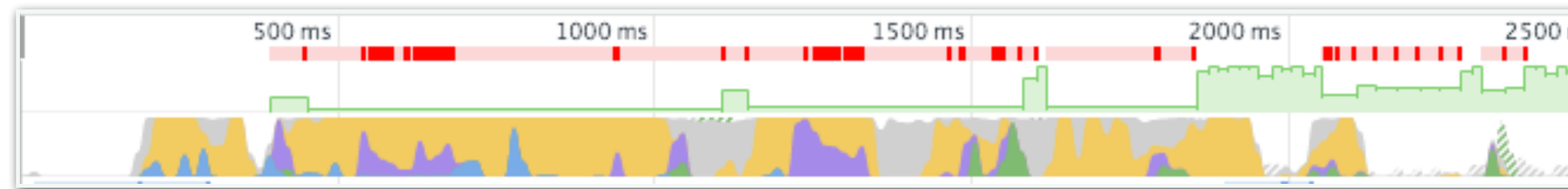
天猫PC端持续消耗内存，重复的rendering



Page rendering



京东M版, 重复的计算DOM的宽高导致大量的layout, 通过chrome 定位发现重复的计算宽度



```
?usid=FVYT3IM5A...n_pcmtiaozhuan index_2016_1_24...jd201604071800 x
i Pretty-print this minified file? more never show x
1 function(){var e=this;var c=$("#body").width();$("#"+e.recommendListId).find("img").forEach(function(g,f){var h=$(g).parent().width();$(g).cs
```

Page rendering



16.66ms



requestAnimationFrame

requestIdleCallback

requestAnimationFrame

兼容代码

复杂运算分段执行，减少丢帧

楼层分模块，通过队列执行

requestIdleCallback

兼容代码

Window.requestIdleCallback(callback, delayTime)

delayTime: 如果没有执行，delayTime后会执行

callback 接收一个deadline 对象作为参数

1: **deadline.timeRemaining()**: 当前帧还剩余多少秒

2: **deadline.didTimeout**: 指定时间是否过期









滚动加载

上报日志，等非需要及时需要操作的代码

不要进行样式或者dom的操作


























page
rendering

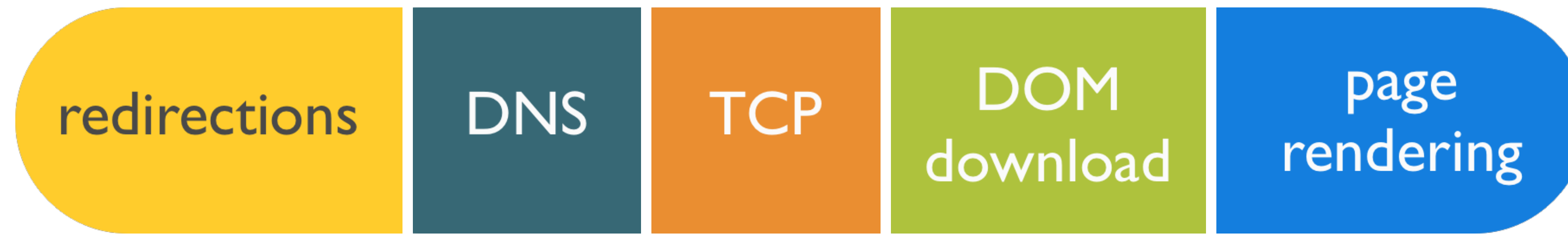
权重： 20%





	redirections	DNS	TCP	Resource	HTTP缓存	WebP	SVG	rendering	
									5
									4
									8
									7



Performance API

无法评估就谈不上优化。幸运的是，Chrome Navigation Timing API 提供了所有必备工具来评估关键呈现路径的每个步骤

performance.now() 比 Date() 更精确, 如422776.420000000004, 从浏览器url打开初始到现在毫秒数

performance.mark(name)

代码中打点

getEntriesByName(name) 通过name获取记时点

getEntriesByType('mark') 获取所有的点数据

measure("difference", "start", "end") 生成新的点 difference: end - start

performance.navigation

redirectCount 同源的重定向

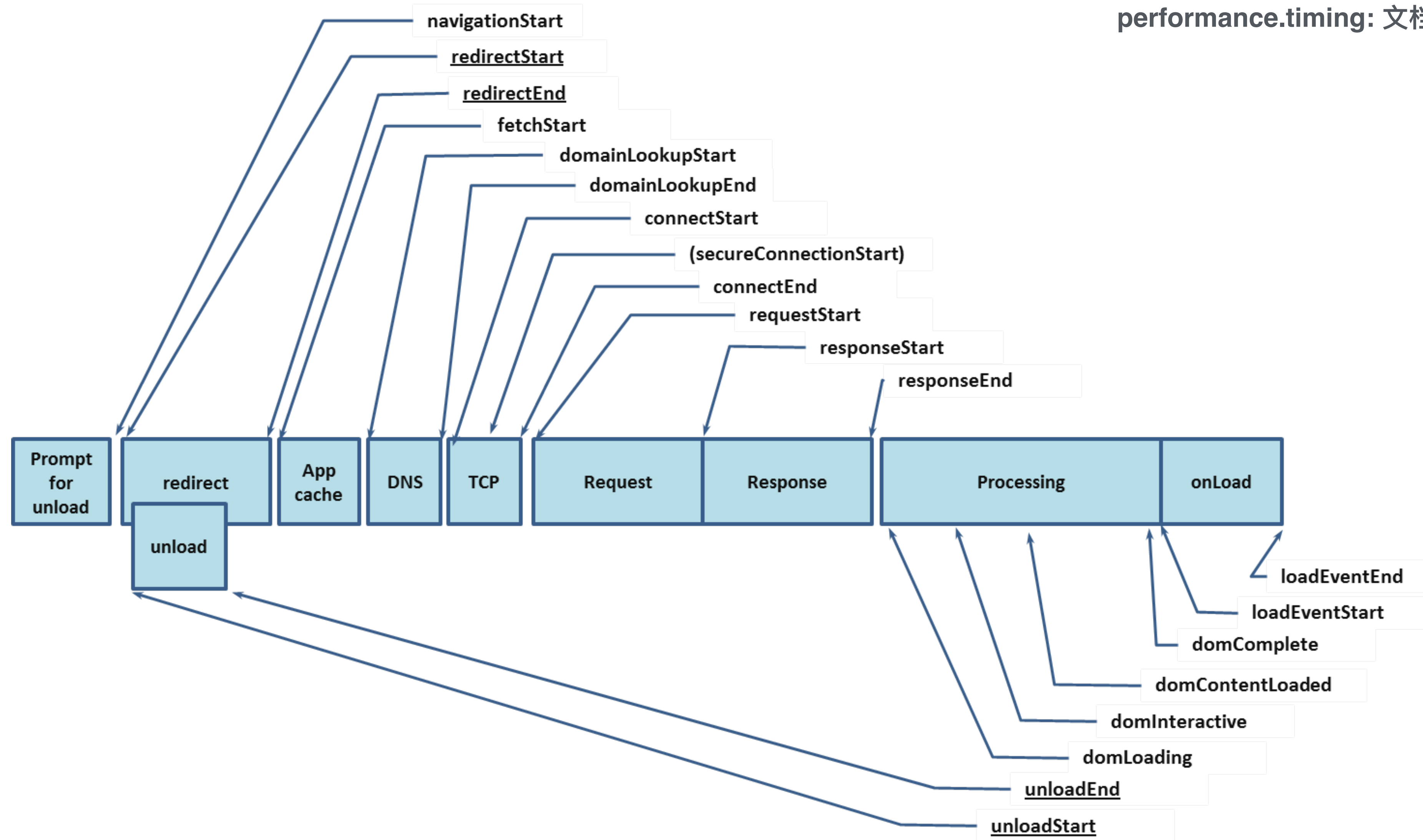
type

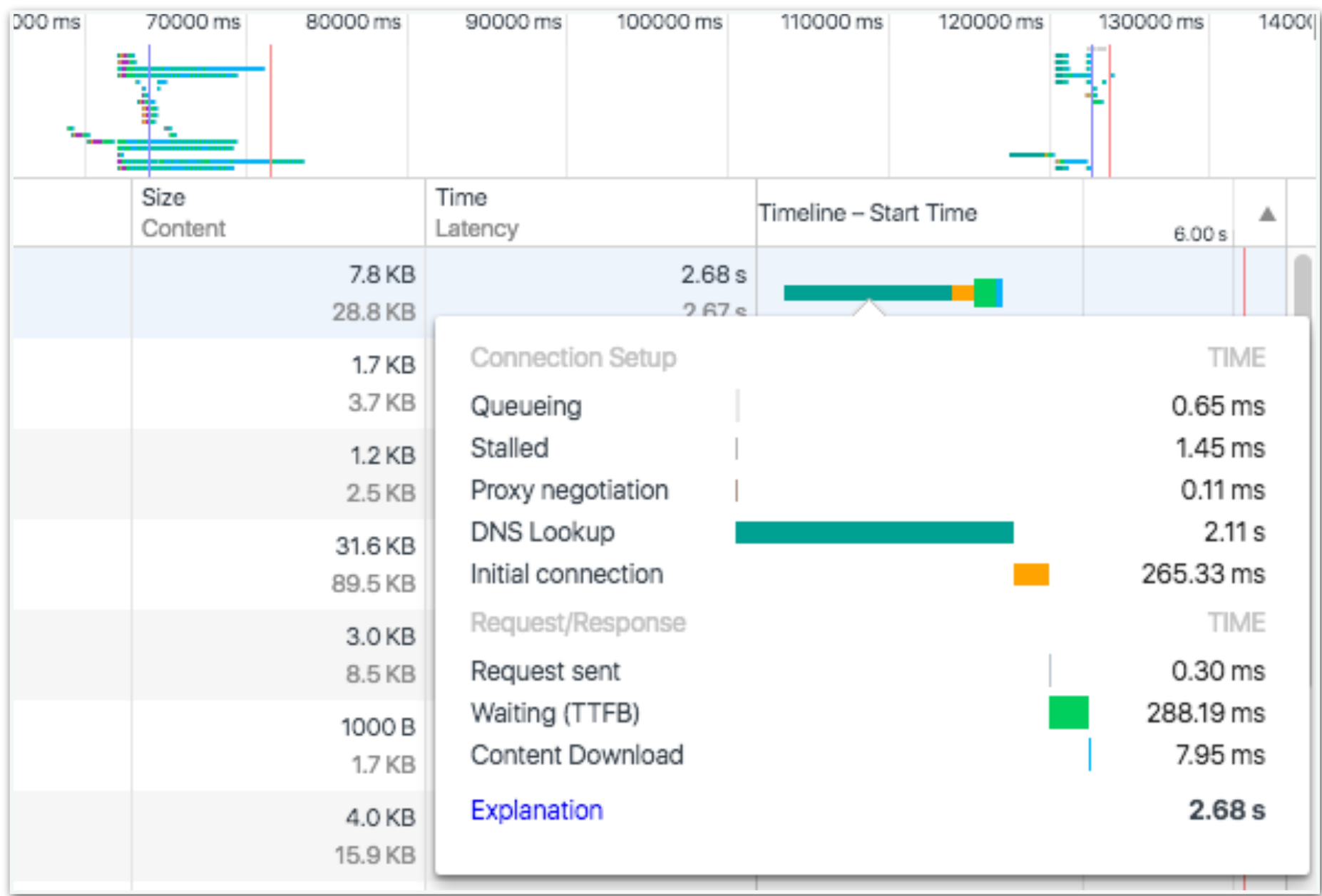
0 : TYPE_NAVIGATE (用户通过常规导航方式访问页面,比如点一个链接, 或者一般的get方式.)

1 : TYPE_RELOAD (用户通过刷新,包括JS调用刷新接口等方式访问页面)

2 : TYPE_BACK_FORWARD (用户通过后退按钮访问本页面)

performance.timing: 文档各个步骤的耗时





Queueing ~ Proxy negotiation : domainLookupStart - fetchStart

DNS Lookup : domainLookupEnd - domainLookupStart

initial connection : connectEnd - connectStart

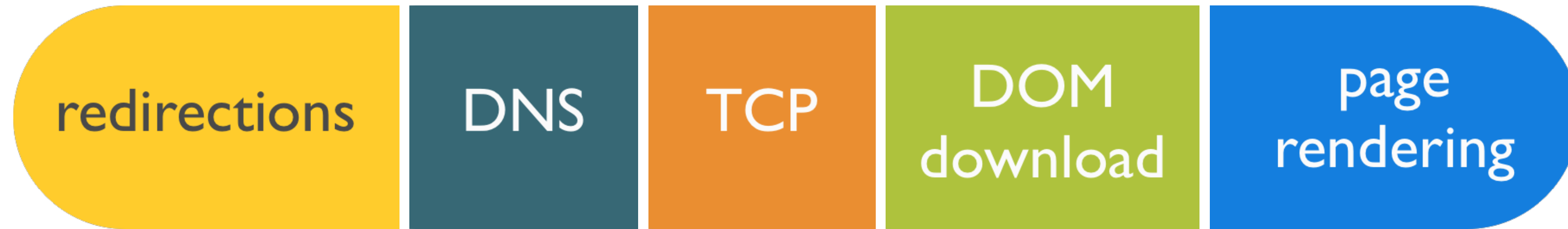
Waiting(TTFB): responseStart - requestStart

Content Download : responseEnd - responseStart

performance.getEntries() : 可以获取页面上所有资源请求状况， 返回PerformanceResourceTiming数组



phantomas



网页的性能优化不止于代码

<http://taobao-fed.org/>

<https://kinsta.com/learn/what-is-http2/>

<http://tools.pingdom.com/>

<https://developers.google.com/speed/pagespeed/?hl=zh-CN>

<https://gtmetrix.com/>

<http://fex.baidu.com/blog/2014/05/build-performance-monitor-in-7-days/>

<https://jakearchibald.com/2016/streams-ftw/>



<https://aotu.io/>

这个PPT的内容过两天回整理成为博文，大家可以关注公众号来订阅

