

首页 / Web前端 / 移动端HTML5<video>视频播放优化实践

移动端HTML5<video>视频播放优化实践

🕒 星期三 2015/01/21 23:23 👤 阅读: 56739 💬 评论: 29 👤 作者: ivan 🏷 分类: Web前端

遇到的挑战

移动端HTML5使用原生<video>标签播放视频，要做到两个基本原则，速度快和体验佳，先来分析一下这两个问题。

■ 下载速度

以一个8s短视频为例，wifi环境下提供的高清视频达到1000kbps，文件大小大约1MB；非wifi环境下提供的低码率视频是500kbps左右，文件大小大约500KB；参考QzoneTouch多普勒测速，2g网络的平均速度是14KB/s，那么下载一个低码率视频耗时35s；那么要想流畅播放视频，就需要一个加载等待的过程，这个过程要有明确的反馈，不能让用户有“坏掉了”的感觉。

多普勒测速数据参考

#	dns(s)	conn(s)	rtt(s)	tran(kb/s)
2g	3.85785	2.33482	2.57478	14.0374
3g	1.60643	0.743109	0.608047	60.1967
wifi	0.986921	0.550208	0.444332	70.8728

■ 用户体验

视频是否可以自动播放，是否能循环播放，是否能显示下载进度，播放的时候如何隐藏控制条，暂停的时候又能显示出来呢。这些问题看上去貌似简单，但是由于PC/iOS/Android这些不同平台、不同的浏览器内核、甚至相同内核的不同版本，所实现的<video>属性、方法和事件差异较大，解决兼容性问题又给开发造成了很大困扰。

分析原因

■ 事件差异

下面是播放一个短视频，在不同平台触发事件和获取属性的差异表现。

PC

#	event	readyState	currentTime(s)	buffered(s)	duration(s)	视频
1	loadstart	NOTHING	0	—	—	—
2	suspend	NOTHING	0	—	—	—
3	play	NOTHING	0	—	—	—
4	waiting	NOTHING	0	—	—	—
5	durationchange	METADATA	0	5.35	7.91	获取到视频
6	loadedmetadata	METADATA	0	0.66	7.91	获取到元数
7	loadeddata	ENOUGHDATA	0	0.66	7.91	—
8	canplay	ENOUGH_DATA	0	0.66	7.91	—
9	playing	ENOUGH_DATA	0	0.66	7.91	开始播放
10	canplaythrough	ENOUGH_DATA	0	0.66	7.91	可以流畅播
11	progress	ENOUGH_DATA	0.11	3.68	7.91	持续下载
12	timeupdate	ENOUGH_DATA	0.14	4.44	7.91	播放进度变
...	...	...	...	...	...	...
23	progress	ENOUGH_DATA	1.77	7.91	7.91	下载完毕
24	suspend	ENOUGH_DATA	1.77	7.91	7.91	—

推荐阅读 Top Read

- 轩枫阁V3.0更新日志
- 前端优化及规范文档
- 浏览简介
- 友情链接
- RSS订阅
- Github

输入关键词搜索相关文章

- 🔥 热门文章 🕒 最新文章 🏷 随机文章
- ① 关于&留言 (901)
 - ② Web前端开发推荐阅读书籍、学... (411)
 - ③ 纸飞机许愿 (184)
 - ④ 关闭Wordpress自动加载的Open ... (109)
 - ⑤ 友情链接 (78)
 - ⑥ 实用的sublime插件集合 - sublim... (72)
 - ⑦ funny (42)
 - ⑧ wordpress自动生成文章目录 (40)
 - ⑨ 多说头像CSS样式自定义-让评论... (33)

新浪微博 Sina Weibo



非常喜欢这张[心] #李宇春流行#

豆瓣 V：【流行】表演者：李宇春，豆瓣评分：7.7，“如同手握一支从容的判笔，将风的轨迹尽数记录。时代从未要求谁，耐心描绘每一朵焰火于夜空消散的过程。但既然年轻气盛，必定温顺尚早。认识你自己，但有限；认识流行，时机正好。”更多专辑信息 <http://t.cn/D1f8kE>

- ### 热门标签 Tag Cloud
- 移动端 调试 js技巧 图片优化
wordpress node jQuery 设计
微信 性能优化 H5 CSS3
Chrome 浏览器 前端 面试 wp
JS 校园招聘 插件 CSS
腾讯招聘 ps 招聘 工具
HTML5 github 技巧 图片 创意
快捷键 js手记

友情链接 Friendly link

- W3CPlus 引领Web前沿
- 站酷 (ZCOOL) - 设计师互动平台
- 前端网 (W3Cfuns) - 前端开发工程师互...
- 优设-UISDC 优秀网页设计联盟
- 新思路团队 New-Thread
- 醉牛前端-前端人的俱乐部
- 前端开发博客

25	timeupdate	ENOUGH_DATA	1.9	7.91	7.91	继续播放中
...	...	...	...	...	...	...
48	timeupdate	ENOUGH_DATA	7.7	7.91	7.91	—
49	timeupdate	ENOUGH_DATA	0	7.91	7.91	—
50	seeking	METADATA	0	7.91	7.91	—
51	waiting	METADATA	0	7.91	7.91	—
52	timeupdate	ENOUGH_DATA	0	7.91	7.91	—
53	seeked	ENOUGH_DATA	0	7.91	7.91	播放完毕进
54	canplay	ENOUGH_DATA	0	7.91	7.91	—
55	playing	ENOUGH_DATA	0	7.91	7.91	循环播放
56	canplaythrough	ENOUGH_DATA	0	7.91	7.91	—
57	timeupdate	ENOUGH_DATA	0.19	7.91	7.91	—
...	...	...	...	...	...	...

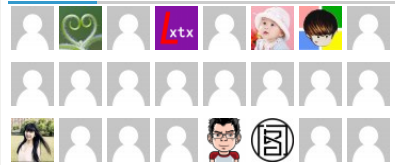
iOS

#	event	readyState	currentTime (s)	buffered (s)	duration (s)	视频
1	loadstart	NOTHING	0	—	—	—
2	play	NOTHING	0	—	—	—
3	waiting	NOTHING	0	—	—	—
4	durationchange	METADATA	0	—	7.91	获取到视频
5	loadedmetadata	METADATA	0	—	7.91	获取到元数
6	loadeddata	ENOUGHDATA	0	—	7.91	—
7	canplay	ENOUGH_DATA	0	7.91	7.91	—
8	canplaythrough	ENOUGH_DATA	0	7.91	7.91	可以流畅播
9	playing	ENOUGH_DATA	0	7.91	7.91	开始播放
10	progress	ENOUGH_DATA	0	7.91	7.91	下载完毕
11	suspend	ENOUGH_DATA	0	7.91	7.91	—
12	timeupdate	ENOUGH_DATA	0.02	7.91	7.91	播放进度变
...	...	...	...	...	...	...
43	timeupdate	ENOUGH_DATA	7.8	7.91	7.91	—
44	timeupdate	ENOUGH_DATA	0	7.91	7.91	—
45	seeked	ENOUGH_DATA	0	7.91	7.91	播放完毕进
46	timeupdate	ENOUGH_DATA	0.22	7.91	7.91	循环播放
...	...	...	...	...	...	...

Android

#	event	readyState	currentTime (s)	buffered (s)	duration (s)	视频
1	loadstart	NOTHING	0	—	—	—
2	play	NOTHING	0	—	—	—
3	waiting	NOTHING	0	0	—	—
4	durationchange	ENOUGH_DATA	0	0	0	—
5	durationchange	ENOUGH_DATA	0	0	7.91	获取到视频
6	loadedmetadata	ENOUGH_DATA	0	0	7.91	获取到元数
7	loadeddata	ENOUGHDATA	0	0	7.91	—

近期访客 Recent Visit



近期评论 Recent Comment

 **xiaohuhu** 03月02日
在 [关于&留言](#) 中评论
可以开放首页banner的源码吗，大佬

 **崔小磊** 02月08日
在 [关于&留言](#) 中评论
请问在移动端怎么用input file调取百度浏览器和搜狗浏览器的视频摄像头

 **影乐** 02月05日
在 [关于&留言](#) 中评论
我本来想进来看文章的，结果我玩了一下午导航栏的钢琴！！

 **piao** 01月25日
在 [关于&留言](#) 中评论
大佬，我想学一下你首页banner的效果

 **许晓霞** 01月16日
在 [关于&留言](#) 中评论
vue中引入怎么报introJs is not defined的错误呢，我确实是引入了

 **大叔汤** 01月12日
在 [关于&留言](#) 中评论
来腾讯之前关注你写的一篇文章，今天偶然看到，喜欢你这个网站，赞一个！

 **飞飞** 01月11日
在 [纸飞机许愿](#) 中评论
【纸飞机许愿】2018加油

 **土家脐哥** 12月26日
在 [纸飞机许愿](#) 中评论
开启

 **Web前端之家** 12月21日
在 [友情链接](#) 中评论
QQ: 1356823313

 **Web前端之家** 12月21日
在 [友情链接](#) 中评论
来做个友情链接吧 我的信息：Web前端之家：www.jiangweishan.com

 **我在** 12月11日
在 [关于&留言](#) 中评论
这个评论框跟不上博客的特效

 **态度xiaomi** 12月11日
在 [关于&留言](#) 中评论
为什么主页是
HTTPd8e87e8d87c694c4dd098b9807e4de6c这里却是https

 **态度xiaomi** 12月11日
在 [关于&留言](#) 中评论
主题很漂亮

 **不忘初衷** 11月28日
在 [funny](#) 中评论
好喜欢你做的网站

 **老蒋博客** 11月01日
在 [轩枫阁VPS升级小记](#) 中评论
朋友 交换链接吗

8	canplay	ENOUGH_DATA	0	0	7.91	–
9	canplaythrough	ENOUGH_DATA	0	0	7.91	–
10	playing	ENOUGH_DATA	0	0	7.91	–
11	timeupdate	ENOUGH_DATA	0	0	7.91	–
12	progress	ENOUGH_DATA	0	3.57	7.91	下载中
13	timeupdate	ENOUGH_DATA	0.2	6.89	7.91	开始播放
14	progress	ENOUGH_DATA	0	7.91	7.91	下载完毕
...	...	...	...	...	...	...
49	timeupdate	ENOUGH_DATA	7.79	7.91	7.91	–
50	progress	ENOUGH_DATA	7.87	7.91	7.91	–
51	timeupdate	ENOUGH_DATA	0	7.91	7.91	–
52	seeking	ENOUGH_DATA	0	7.91	7.91	播放完毕进
53	timeupdate	ENOUGH_DATA	0	7.91	7.91	–
54	seeked	ENOUGH_DATA	0	7.91	7.91	循环播放失
55	progress	ENOUGH_DATA	0	7.91	7.91	–
56	stalled	ENOUGH_DATA	0	7.91	7.91	–

一些常用且需要重点关注的<video>事件

event	iOS	Android
*****	*****	*****
play	只是要播放视频，响应的是video.play()方法，并不代表已经开始播放	和iOS一样，仅是响应video.play()方法
durationchange	会执行一次，一定会获取到视频的duration	可能会执行多次，只有最后一次才能获取到真实的duration前面的duration都是0；但低版本Android可能获取到的duration:（本文提到的低版本Android大部分是4.
canplay	可以认为是视频元素没有问题，可以运行，没有更多含义了，基本用不上	同iOS
canplaythrough	会有明确的缓冲，表示可以流畅播放了；	没有什么用，视频仍然会卡住，数据可能还没有开始加载；
playing	明确表示播放开始了；	依然没有用，视频可能并没有开始播放
progress	有明确的下载，可以获取到当前的buffer，并且全部下载完毕后不在触发；	不一定有明确的数据下载，并且全部下载完毕后依然继续触发；
timeupdate	会有明确的进度变化，可以获取到currentTime；	进度不一定变化，currentTime可能总是但是第一次有currentTime变化的timeup
error	iOS中会有明确的错误抛出；	Android中某些浏览器会莫名其妙的抛出
stalled	网络状况不佳，导致视频下载中断；	在没有play之前，也可能会抛出该事件。

■ 属性差异

attributes	iOS	android
*****	*****	*****
poster 封面图片	支持，但是加载速度明显比在中要慢；	不一定支持（浏览器厂商的实现标准不统一）；
preload 预加载	iPhone不支持；	可能支持；
autoplay 自动播放	iPhone Safari中不支持，但在webview中可能被开启； iOS开发文档明确说明蜂窝网络下不允许autoplay；	可能支持；

loop 循环播放	支持	可能支持；
controls 控制条	支持，但是需要开始播放了才显示	基本都支持显示或者不显示
width和height	一定给出明确的属性设置，切不能为0；	如果不设置， 仅仅通过CSS样式去控制视频大小， 可能会导致标签失效。

■ 其他怪异bug和不好表现

ios	android
*****	*****
物理位置覆盖在<video>区域上的元素， click和touch等事件会失效，比如一个<a> 链接如果覆盖在<video>上， 那么点击后没有任何效果。	—
iOS8.0+中，单页面播放视频超过16个， 再播放的视频全部MediaError解码异常无法播放。	—
iPhone的Safari会弹出一个全屏的播放器来播放视频， iPad则支持内联播放。iOS7+ 如果webview（比如微信） 开启了 webview.allowsInlineMediaPlayback = YES； 可以通过设置 webkit-playsinline 属性支持内联播放；	支持内联播放， 但某些厂商会用自己的播放器劫持原生的视频播
下载视频时， 会先发送一个2字节的请求来获取视频元数据 （比如时长），然后再不断的发送分包续传（206） 请求来下载视频， 抓包显示请求数和请求量至少有一倍的冗余（x2）， 这个严重的bug在iOS8中有明显的修复， 但是分包的206请求仍然会有冗余数据的下载， 浪费了流量。	比iOS的处理方式好，没有第一个2字节请求， 没有流量损耗；
—	低版本Android（<=4.0.4）中，<video> 如果有相对和绝对定位的层中， 可能会导致整个页面错位。
—	某些浏览器厂商会劫持<video>， 用其“自己”的播放器来播放视频，“破坏”了产品， 那么只能case by case的解决了。
加载视频时没有进度提示， 视觉上看不出是播放完了还是卡住了；	加载视频时，大都会显示一个自带的loading UI（菊花）。

最佳实践

■ 视频初始化

如果将一个<video>直接显示在页面中，那么就会看到各种五花八门的播放器初始效果；



这显然不是一个好的视觉体验，那么通常的做法是制作一个模拟的视频播放视图，比如一个封面加一个播放

按钮。

而真实的<video>视频元素要隐藏起来，如何隐藏呢？最好不要用 {display: none} 或者 {width:0;height:0;} 的方式，因为这样视频元素会处于未激活的状态，给后续的处理带来麻烦。最佳的方式是将视频设置成1×1像素大小，放在视觉边缘的位置。

```
1 <!--iOS-->
2 <video webkit-playsinline width="1" height="1" class="vplayinside notaplink" x-webkit-
3
4 <!--Android-->
5 <video width="1" height="1" controls loop="loop" src="%=src%"></video>
```



■ 自动播放

autoplay的支持依赖内核和网络状况，比如iPhone在蜂窝网络下明确禁用了autoplay；

经过试验，在没有明确的用户操作的情况下，直接通过 video.play() 也是无法激活播放的；

并且在产品设计上，自动播放也不是一个舒服的用户体验，所以产品设计上尽量避免使用自动播放。

■ 点击播放

之前提到，视频最好通过1px大小隐藏起来，那么这时如何触发播放呢？

经过试验，当在明确的用户操作（touch、click）时，通过这些用户行为事件的回调函数，用 video.play() 是可以触发视频播放的，那么能否在用户操作后，再去同步的创建和播放视频呢？答案是肯定的，这无疑是一个视频元素初始化的最佳实践，但是有些差异需要注意。

iOS6+

可以在用户的touch时间中动态创建并播放视频。

iOS < 6

可以在用户的touch时间中动态创建视频，但不能播放；要再追加一个click事件来启动播放；也就是说，给伪造的视频播放按钮同时绑定tap和click事件，在tap的时候创建，在之后300毫秒的click中去播放。

Android

大部分高版本Android可以像iOS6+那样去处理，但是低版本的不行，必须要通过click事件去传递 video.play()，为了保持兼容，最好是用帮tap和click两个事件来分别完成视频的初始化和播放。

我们还发现，有些低版本Android中，无法通过 video.play() 来播放视频，必须有真实的用户点击视频元素才能播放；这种情况，有一个技巧就是在tap的时候初始化并放大视频覆盖在播放视图中，让300毫秒后的真实点击行为穿透点击在视频元素上来实现播放。

■ 循环播放

如果视频需要循环播放，那么就增加 loop 属性，是否能循环播放就看浏览器是否支持了，因为还没有找到 hack技巧来强制循环播放；

即使，在不支持循环播放的Android中，通过监听 seeked 事件知道了播放进度到了终点或起点暂停了，此时也无法通过 video.play() 来让视频重新播放。

■ 监控下载进度

如何获取视频时长和已经下载的时长？

```
1 // 视频时长
2 var duration = video.duration
3
4 // 获取视频已经下载的时长
5 function getEnd(video) {
```

```

7   try {
8       end = video.buffered.end(0) || 0
9       end = parseInt(end * 1000 + 1) / 1000
10  } catch(e) {
11  }
12  return end
13 }

```

progress事件表示视频在加载，但是它的触发频率和时机并不规律，最佳做法是通过一个定时器去实时获取end，当end >= duration时，表示已经下载完毕，再终止定时器。

```

1  var timer = setInterval(function() {
2      var end = getEnd(video),
3          duration = video.duration
4
5      if(end < duration) {
6          return
7      }
8
9      clearInterval(timer)
10 }, 1000)

```

■ 全部下载后再播放

假设播放短视频，如果网络不佳，会造成播放断断续续，在iOS中这种停顿还没有一个明确的等待提示，这不是一个好的体验，那么是否可以将视频全部下载完毕再播放呢？

在iOS中，可以在视频刚开始下载的时候马上暂停，此时下载还将继续，可以做一个loading的菊花告知视频正在加载，然后等到视频全部下载完再开始播放。

```

1  $(video).one('loadeddata', function() {
2      // 暂停，但下载还在继续
3      video.pause()
4
5      // 启动定时器检测视频下载进度
6      var timer = setInterval(function() {
7          var end = getEnd(video),
8              duration = video.duration
9
10         if(end < duration) {
11             return
12         }
13
14         var width = $(video).parent().width()
15
16         // 下载完了，开始播放吧
17         $(video).attr({
18             width: width,
19             height: width
20         })
21         video.play()
22
23         clearInterval(timer)
24     }, 1000)
25 })

```

■ 缓冲播放——边下边播时，选择开始播放的最佳时间点

当视频越来越长或者网络慢时，等待视频全部下载完再播放也不是好的体验，最好能边下边播，缓冲到流畅状态就开始播放，那什么时候播放才是最佳时间点呢？

在iOS中，canplaythrough事件就是这个最佳时间点，它是通过动态计算缓冲量和下载速度得出的视频可以流畅播放的状态反馈。

canplaythrough event: The user agent estimates that if playback were to be started now, the media resource could be rendered at the current playback rate all the way to its end without having to stop for further buffering.

注意：下载完再播放和缓冲播放只适用于iOS。

■ 统计播放时间和播放次数

要统计实际的播放时间，要累加timeupdate事件变化的时间，再减去中间可能暂停的时间。

```
1 $video.on('playing', function() {
2   // 开始播放是打点
3   $video.attr('data-updateTime', +new Date())
4 })
5
6 $video.on('pause', function() {
7   // 暂停播放时清除打点
8   $video.removeAttr('data-updateTime')
9 })
10
11 // 累加播放时间
12 $video.on('timeupdate', function(event) {
13   var $video = $(event.target),
14       updateTime = parseInt($video.attr('data-updateTime') || 0),
15       playingTime = parseInt($video.attr('data-playingTime') || 0),
16       times = parseInt($video.attr('data-times') || 0),
17       newtimes = 0,
18       video = $video.get(0),
19       duration = parseFloat($video.attr('data-duration') || 0),
20       now = +new Date()
21
22   // 播放时间
23   playingTime = playingTime + now - updateTime
24
25   // 播放次数
26   newtimes = Math.ceil(playingTime / 1000 / duration)
27 })
```

```
28 $video.attr('data-playingTime', playingTime);
29 $video.attr('data-updateTime', now);
30 })
```

异常处理

对error事件做详细的上报;
对stalled事件做统计上报, 并提示用户网络慢等。

参考数据

微视触屏版iOS视频测速

网络环境	视频码率	获取到视频时长 时间点(s)	开始流畅播放 时间点(s)	全部下载完毕 时间点(s)	视频长度(s)
wifi	1000kbps	2.86	3.97	5.85	8.69
非wifi	500kbps	4.56	8	10.62	8.67

参考资料

- HTML5 Video Events and API检测工具 <http://www.w3.org/2010/05/video/mediaevents.html>
- W3C video 标准 <http://www.w3.org/TR/html5/embedded-content-0.html#the-video-element>
- 如何在iOS7+的webview中内联播放视频 <http://darktalker.com/2014/play-video-inline-iphone-ios7>
- 视频事件流水查看工具 <http://z.weishi.qq.com/app/video.html>

参考自: [zhaoda](#)

标签: #HTML5、#video、#移动端、#视频

分享到: 

本文标题: [移动端HTML5<video>视频播放优化实践 - 轩枫阁](#)
转载请务必注明出处, 欢迎分享

如果觉得我的文章对您有用, 请随意打赏

赞 赏

相关文章



piano-play 基于html5的钢琴导航弹奏游戏



移动端开发: iOS与Android平台上问题列表



移动端开发工具类方法汇总



wordpress移动端自动切换主题代码