

Android的WebView控件载入网页显示速度慢的终极解决方案

Android的WebView控件载入网页显示速度慢的终极解决方案

【转载来源自<http://hi.baidu.com/goldchocobo/>】

Android客户端中混搭HTML页面，会出现虽然HTML内容载入完成，标题也正常显示，但是整个网页需要等到近5秒（甚至更多）时间才会显示出来。研究了很久，搜遍了国外很多网站，也看过PhoneGap的代码，一直无解。

一般人堆WebView的加速，都是建议先用webView.getSettings().setBlockNetworkImage(true); 将图片下载阻塞，然后在浏览器的OnPageFinished事件中设置webView.getSettings().setBlockNetworkImage(false); 通过图片的延迟载入，让网页能更快地显示。

但是，通过实际的日志发现，Android的OnPageFinished事件会在Javascript脚本执行完成之后才会触发。如果在页面中使用JQuery，会在处理完DOM对象，执行完\$(document).ready(function() {});事件自会后才会渲染并显示页面。如下图

01-04 07:59:58.702	D	800	ZhuiDebug	EndLoadingHtm
01-04 07:59:58.901	D	800	ZhuiDebug	onPageStarted:http://a
01-04 07:59:59.681	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.721	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.731	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.751	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.751	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.771	D	800	ZhuiDebug	onLoadResource:http://
01-04 07:59:59.771	D	800	ZhuiDebug	onLoadResource:http://
01-04 08:00:07.562	D	800	ZhuiDebug	onLoadResource:http://
01-04 08:00:07.602	D	800	ZhuiDebug	onLoadResource:http://
01-04 08:00:10.282	D	800	ZhuiDebug	onPageFinished
01-04 08:00:10.702	D	800	ZhuiDebug	onLoadResource:http://
01-04 08:00:10.702	D	800	ZhuiDebug	onLoadResource:http://
01-04 08:00:11.491	D	800	ZhuiDebug	onLoadResource:http://

可以看到在载入完最后一个JS脚本之后，对DOM元素的渲染和处理就花了8秒，然后执行了AJAX方法载入外部页面又花了2、3秒，最后才会触发onPageFinished显示页面。再往后由于程序中设置了setBlockNetworkImage(false)，所以开始载入外部图片。（如果不控制这个参数，图片载入会在渲染期间下载。综上，由于JS脚本的处理，造成了一张页面打开多花了10秒左右时间。而同样的页面在iPhone上却是载入相当的快，显示完页面才会触发脚本的执行。（提示：如果使用JQueryMobile，更会慢得离谱）

要解决这个问题，就是想办法让浏览器延迟加载JS脚本，但是Android的WebView控件没有这样的参数。无法单独阻塞JS脚本，另外有个setBlockNetworkLoads，用了之后也无法实现类似图片的异步载入的功能，页面成了光板，连CSS也阻塞了。

就是这个问题困扰了很久，直到在做HTML照片墙时，由于setBlockNetworkImage在OnPageFinished之后才会释放，导致在JS脚本载入图片过程中无法获取图片的高宽信息，最后巧妙地通过\$(document).ready(function(){setTimeout(func,10)});，成功将函数在onPageFinished之后运行。那么延伸来想，是否可以将JS脚本也用同样的方式延迟载入呢？

答案是肯定的，在<http://wonko.com/post/painless-javascript-lazy-loading-with-lazyload>可以找到JS脚本延迟加载的第三方组件。

我改造了之前速度奇慢的界面，我所使用的核心JS代码如下：

```
<script src="/css/j/lazyload-min.js" type="text/javascript"></script>
```

```
<script type="text/javascript" charset="utf-8">
```

```
loadComplete(){
```

```
    //instead of document.read()
```

```
}
```

```
function loadscript()
```

```
{
```

```
LazyLoad.loadOnce([
```

```
    '/css/j/jquery-1.6.2.min.js',
```

```
'/css/j/flow/jquery.flow.1.1.min.js',  
'/css/j/min.js?v=2011100852'  
], loadComplete);  
  
}  
  
setTimeout(loadscript,10);  
  
</script>
```

就是混搭setTimeout和layzload，让JS脚本可以真正在onPageFinish之后执行。

最终执行的效果如图：

01-04 08:41:12.071	D	1800	ZhuiDebug	EndLoadingHtal	
01-04 08:41:12.701	D	1800	ZhuiDebug	onPageStarted:http://api.:	
01-04 08:41:13.411	D	1800	ZhuiDebug	onLoadResource:http://api	'css/c/tb.css?v=20111206
01-04 08:41:13.411	D	1800	ZhuiDebug	onLoadResource:http://api	'css/c/skin.css?v=20111206
01-04 08:41:14.322	D	1800	ZhuiDebug	onLoadResource:http://api	'css/j/layzload-min.js
01-04 08:41:14.351	D	1800	ZhuiDebug	onPageFinished	
01-04 08:41:14.521	D	1800	ZhuiDebug	onLoadResource:http://api	'css/j/jquery-1.6.2.min.js
01-04 08:41:14.532	D	1800	ZhuiDebug	onLoadResource:http://a.tl	'w/app/trade/hi/images/icon-slide-btn.png
01-04 08:41:14.532	D	1800	ZhuiDebug	onLoadResource:http://api	'css/i/gobuy.png
01-04 08:41:14.551	D	1800	ZhuiDebug	onLoadResource:http://api	'utility/imageshow.ashx?LimitWidth=220&u=aHR0c
01-04 08:41:14.551	D	1800	ZhuiDebug	onLoadResource:http://api	'css/images/wangvang.png

可以看到非常显著的改善，从onPageStarted到onPageFinished只用了2秒不到的时间，这个时间主要花在HTML和CSS渲染上。从界面上来看，就是一闪而过的网页载入进度条，立即显示CSS渲染过的页面效果，然后再载入并执行JS脚本，逐块显示需要通过AJAX获取的数据。

综上，解决Android载入WebView页面慢的问题，不是Android程序员的事情，而是Web前端工程师的问题。如果您使用基于WebView的Android第三方壳工具（例如PhoneGap，可以通过这种方式改善UI界面的响应时间）。

发布这个解决方案，希望基于Web的客户端解决方案能更上一层楼，让HTML和原生APP混搭或的纯WEBAPP实现效果可以更理想，功德无量.....