解决页面使用overflow: scroll在iOS上滑动卡顿的问题

一：Lying人生感悟(可忽略)

摩西奶奶曾经说过：世界上，最公平和最不公平的，都是时间。别人偷不走它。而你也留不住它。你拥有它，却不能改变它。光阴里的艰难或是快乐，它都一一带走。身处其中的你我，年轻或是衰老，所能做的，就是充分去享用它，享受每一个生命时期，收藏每一个年龄段带给你的感动与美好。

是的，这句话我也曾经在某个令人刻骨铭心的夜晚对一个人说过。过去了的事情再提及难免让人有些怀念。既然走不出来又何必再强求自己走出来。所以让该来的到来，让不该走的别走，珍惜所拥有的一切期许是最好的选择.....

二：回到正题(待细看)

故事背景：最近的一次开发中，使用到了overflow：scroll 属性来滑动div。信心满满的以为不会出现任何问题，看来还是太清高自傲了，于是写下这篇随笔特此总结一番。

如果你对某个div或模块使用了overflow: scroll属性，在iOS系统的手机上浏览时，则会出现明显的卡顿现象。但是在android系统的手机上则不会出现该问题。大家不妨可以分别使用IOS和Android系统的手机浏览以下链接或扫描二维码后滑动文字区域查看该效果(重点是记住iPhone浏览时的效果，方便浏览后文)：http://geek100.com/demo/os.html.

表示很奇怪会产生这样的差异，于是卸下行囊、放下面包、拿出电脑、插上网线、双击chrome、在输入栏中默默地敲上baidu.com...大百度(我更习惯性的称作为大败毒，因为它几乎构成了我学习、工作、生活的全部)。通过一个早上的爬虫搜索和与前端开发高手的技术探讨得知以下代码可解决这种卡顿的问题：

```
-webkit-overflow-scrolling: touch;
```

嗯，这次收拾好心情、重拾之前唾手可弃的信心(不是节操哦)、利用PP助手谨小慎微地把文件拖入iPhone手机中、小心翼翼地点开页面、手指轻轻在屏幕上滑动...哇哦，果然滑动流畅了诶。来吧！朋友!不妨拿出IPhone一起感受这激动的时刻。链接：http://geek100.com/demo/ost.html

据说是因为这行代码启用了硬件加速特性，所以滑动很流畅。但是这个属性也会相对耗费更多内存。在流畅的滑动效果和耗费内存之间，我选择了前者。

后来深入研究了一下该属性。具体深入点如下：

实际上，Safari真的用了原生控件来实现，对于有-webkit-overflow-scrolling的网页，会创建一个UIScrollView，提供子layer给渲染模块使用。创建时的堆栈如下：

Thread 1, Queue : com.apple.main-thread #0 0x00086723 in -[UIScrollView initWithFrame:] () #1 0x004ec3bd in -[UIWebOverflowScrollView initWithLayer:node:webDocumentView:] () #2 0x001f1769 in -[UIWebDocumentView webView:didCreateOrUpdateScrollingLayer:withContentsLayer:scrollSize:forNode:allowHorizontalScrollbar:allowVerticalScrollbar:] () #3 0x01d571bd in __invoking___ () #4 0x01d570d6 in -[NSInvocation invoke] () #5 0x01d5724a in -[NSInvocation invokeWithTarget:] () #6 0x027fb6a1 in -[_WebSafeForwarder forwardInvocation:] () #7 0x027fb8ab in __44-[_WebSafeAsyncForwarder forwardInvocation:]_block_invoke_0 () #8 0x04ac753f in _dispatch_call_block_and_release () #9 0x04ad9014 in _dispatch_client_callout () #10 0x04ac97d5 in _dispatch_main_queue_callback_4CF () #11 0x01d09af5 in

__CFRunLoopRun () #12 0x01d08f44 in CFRunLoopRunSpecific () #13 0x01d08e1b in CFRunLoopRunInMode () #14 0x01cbd7e3 in GSEventRunModal () #15 0x01cbd668 in GSEventRun () #16 0x00032ffc in UIApplicationMain () #17 0x00002ae2 in main at /Users/liuhx/Desktop/UIWebView_Research/WebViewResearch/main.mm:16

实际创建的是UIWebOverflowScrollView，它继承自UIScrollView，声明为：

@class DOMNode, UIWebDocumentView, UIWebOverflowContentView, UIWebOverflowScrollListener; @interface UIWebOverflowScrollView : UIScrollView { UIWebDocumentView *_webDocumentView; UIWebOverflowScrollListener *_scrollListener; UIWebOverflowContentView *_overflowContentView; DOMNode *_node; BOOL _beingRemoved; } @property(nonatomic, getter=isBeingRemoved) BOOL beingRemoved; // @synthesize beingRemoved=_beingRemoved; @property(retain, nonatomic) DOMNode *node; // @synthesize node=_node; @property(retain, nonatomic) UIWebOverflowContentView *overflowContentView; // @synthesize overflowContentView=_overflowContentView; @property(retain, nonatomic) UIWebOverflowScrollListener *scrollListener; // @synthesize scrollListener=_scrollListener; @property(nonatomic) UIWebDocumentView *webDocumentView; // @synthesize webDocumentView=_webDocumentView; - (void)setContentOffset: (struct CGPoint)arg1; - (void)_replaceLayer:(id)arg1; - (void)prepareForRemoval; - (void)fixUpViewAfterInsertion; - (id)superview; - (void)dealloc; - (id)initWithLayer:(id)arg1 node:(id)arg2 webDocumentView:(id)arg3; @end

其还有一个子View作为ContentView，是给WebCore真正用作渲染overflow型内容的layer的容器。UIWebOverflowContentView的声明为：

@interface UIWebOverflowContentView : UIView { } - (void)_setCachedSubviews:(id)arg1; - (void)_replaceLayer:(id)arg1; - (void)fixUpViewAfterInsertion; - (id)superview; - (id)initWithLayer:

(id)arg1; @end

再往底层跟，都是CALayer的操作。以上两个类都是UIKit层的实现，需要WebCore有硬件加速的支持才有实际意义，相关的逻辑被包含在ACCELERATED_COMPOSITING这个宏里。

原理说了一大堆，我表示一句也没看明白。不过呢，作为知识的分享者就应该要时时刻刻以最简单明了的说法阐述问题，所以总结以下几点供大家参考：

1.
    1. 从SVN log看，在WebKit 108400版本左右才支持，所以iOS Safari应该是需要5.0。Android则是在4.0以上支持。
    2. 从前端开发的角度讲，只需要知道CSS的属性-webkit-overflow-scrolling是真的创建了带有硬件加速的系统级控件，所以效率很高。
    3. 从实际开发的角度讲，采用这样的做法相对是耗更多内存的，最好是在产生了非常大面积的overflow时才应用。

三：补充内容(待注意)

1. 上述所说的方法的确可以解决ios5.0、android4.0以后系统的滑动卡顿问题，不过呢在这还可以为大家推荐一些相关插件：iScroll(这里是iScroll插件的中文地址)、jRoll(中文名：酸萝卜)。
2. 关于掌握更多的解决方案或者经验的朋友不妨在评论下方留言吧。关于阅读本篇文章的读者也欢迎你们在评论版里对文章以及本人做出及时的评价与建议。让我们共同奋进！！！在这向你们表示感谢。
3. 在这向文中所提到的前端高人表示真诚感谢。谢谢前辈的耐心解答以及给予的宝贵经验。同时也感谢前辈在我工作之余的时间里化作暗夜中的一盏明灯，照亮我前行的方向。
4. 在这向大家保证该随笔中的任何一条链接都经过严格的筛选和把关。100%保证无毒无公害。用伊利纯牛奶的广告语改编后来说则是：百次验证信赖，见证链接品质!(广告：本随笔由好喝的不要不要的伊利纯牛奶赞助支持.....链接：https://yili.tmall.com/search.htm)