

Iframe 加载技术和性能问题

blog.csdn.net

2012年09月11日 21:44

前提话：

搞性能的人都知道高性能网站建设指南的作者steve讲了很多怎么处理代码，使你的网站速度很快。也包括讲了Iframe 的 blocking问题，今天在网上看了Aaron的关于讲Iframe的性能提出了新的思路。好的东东要分享，决定做个翻译（虽然英文很差但好东东就要说出来）。在性能方面的东东国内确实讲得很少。

Iframe setTimeout() does not work in IE9

Chad Barnsdale of

Unfinishedman.com

mentioned to me on May 25 that the Iframe setTimeout() technique does not work in IE9. And he was right. The file that gets loaded into the iframe simply does not load. Nothing happens. I will dive into this sometime soon. It's probably something small with the JavaScript code.

在IE9中使用Iframe setTimeout()的方式不能工作。注释一下吧。在steve中讲了iFrame不阻塞时讲到可以使用Iframe setTimeout()可以解除阻塞，为了能了解setTimeout实现方式。我这里贴两段代码：

1.最为普通的iframe代码，blocking page

```
<iframe id=iframe1 src="iframe-empty.php?t=" width=95% height=150 border=2> </iframe>
```

Demo:

<http://stevesouders.com/efws/iframe-onload-blocking.php?t=1347283666>

2.steve的setTimeout实现iframe代码

```
<iframe id=iframe1 src="" width=95% height=150 border=2> </iframe>
```

```
<script type="text/javascript"> function setIframeSrc() {   iframe1 = document.getElementById('iframe1'); if (
iframe1 ) {   if ( -1 == navigator.userAgent.indexOf("MSIE") ) {       t_iframe = Number(new Date());   iframe1.src =
"iframe-empty.php?t=";   }   else {       t_iframe = Number(new Date());   iframe1.location = "iframe-empty.php?t=";   }
} else {   alert("ERROR: There was a race condition - the iframe doesn't exist.");   } }
```

```
etTimeout(setIframeSrc, 5); </script>
```

Demo:

<http://stevesouders.com/efws/iframe-onload-nonblocking.php?t=1347283111>

大家明白了吧，其实原理很简单因为iframe会blocking页面，为了解决blocking,让这段iframe晚执行即你在设定的秒数后再执行。

前段话的大意是：5.25日Unfinsehman.com网站的Chad Bransdale向我提iframe的setTimeout技术不能在IE9工作了，他说的是对的，加载到iframe的文件不加载了。什么也没有发生，这时我很快的研究了下，这可能是javascript的问题。

Iframes are often used to load third party content, ads and widgets. The main reason to use the iframe technique is that the iframe content can load in parallel with the main page: it doesn't block the main page. Loading content in an iframe does however have two downsides, as Steve Souders outlines in his blog post

Using Iframes Sparignly

Iframes block onload of the main page The main page and iframe share the same connection pool

Iframe被经常用于加载第三方内容，比如广告和其它小玩意，主要原因是使用iframe技术是iframe能在主页面中并行加载，它不阻止主页在，然而在iframe中加载内容有两个缺点的地方，正如steve souders在他的blog提到的保守的使用iframe.大家可以通过链接去看看，

iframe阻塞主页面加载。

主页面和iframe共享同一个连接池。

备注：关于steve souders讲述的using iframe sparignly见翻译

<http://www.99css.com/archives/36>

The onload blocking is the biggest problem of the two and hurts performance the most. You really want the load event to fire as soon as possible, for optimal user experience of course but also because of how Google measures your site speed: Google Toolbar in IE and FF browsers of site visitors measures time to onload.

加载阻塞是两个缺点中最大的问题并且大多数最影响性能。你真正的想尽快的加载事件到fire,给用户最佳的用户过程并且如何用google测量你网站速度：在网站访问者的IE和Firefox浏览器中的google工具条测量加载时间

How can you load an iframe without blocking onload and as a result improve page performance?

怎么样无阻塞的加载iframe，作为一个结果提高网页性能的结果

This article shows 4 ways to load an iframe in a web page:

Normal Iframe

Iframe After Onload

Iframe setTimeout()

and

Dynamic Asynch Iframe

. For each the behaviour related to the load event is described and shown by way of IE8 waterfall charts. I recommend to look closely at the Dynamic Asynch Iframe technique because this one is great for performance. Oh, and as a bonus, I throw in the

Friendly Iframe

technique. It doesn't really qualify as an iframe

loading

technique, but it has to do with iframes, ads and (non)-blocking.

这篇文章讲述了一个网页中加载iFrame的四个方法：普通加载Iframe,在onload以后加载Iframe,Iframe setTimeout 和动态同步Iframe.每个方法都在IE8测试过，并用瀑布图表示。我推荐动态加载iframe技术因为他的性能比较好，作为奖励，我投入了Friendly iframe技术研究，他不是真正合适的作为iframe加载技术，但是它没有阻塞的处理iframe,广告

You all know this one. It's the default way to load an iframe and works in all browsers: `<iframe src="/path/to/file" frameborder="0" width="728" height="90" scrolling="auto"></iframe>`

普通加载iframe

众所周知，这是缺省的加载iframe方式，工作在所有的浏览器中

`<iframe src="/path/to/file" frameborder="0" width="728" height="90" scrolling="auto"></iframe>`

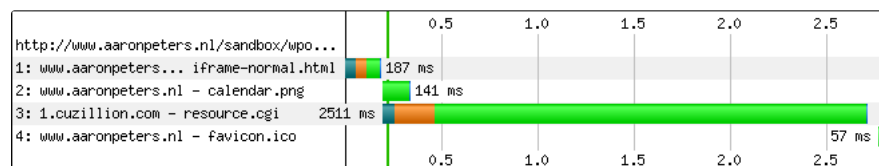
Using the Normal Iframe technique will result in the following behaviour in all browsers:

使用普通的iframe技术将得到下面的在所有浏览器中的表现：

Iframe在主页面的onload之前加载 iframe会在所有的iframe的内容加载完成后触发iframe的onload事件 iframonload会在iframe onload完成之后再触发，iframe阻塞了主页的加载

当iframe加载期间，一个或多个浏览指示器显示忙表示一些事在加载

I created a simple test page and ran it through Webpagetest.org (IE7 and IE8). The waterfall chart below clearly shows how the Normal Iframe blocks the main page onload.



Normal Iframe - Performance - IE8 waterfall chart

我创建了一个简单的测试页面并且在webpagetest.org(IE7和IE8)运行，这个瀑布图清晰的显示普通的iframe如何不过阻塞了主页面的加载。

My advice: be aware of the onload blocking. It's not a big problem if the iframe content takes a short time to load (and execute), and the iframe in itself is good to use because it loads in parallel with the main page. But if it takes long for the iframe to finish, the user experience is damaged. Test it on your page(s) and run it through Webpagetest.org a few times, look at the impact on onload (Doc Complete time) and decide if you need a better technique.

我的建议：意识到加载阻塞，这不是很大的问题如果iframe内容花很短时间能加载，并且iframe可以善于使用因为他可以和主页面并行加载。但是如果要花长时间加载iframe，用户的体验被损坏了。在webpagetest.org上花几分钟测试你的页面，观察doc complete time的加载影响然后决定是否你需要更好的技术。

Imagine you want to load something in an iframe, but it's not really important for the page. Or the content of the iframe is not immediately visible to the user because it's way down below the fold or hidden behind a link/tab. Consider deferring the loading of the iframe until after the main page is done.

在onload之后加载iframe

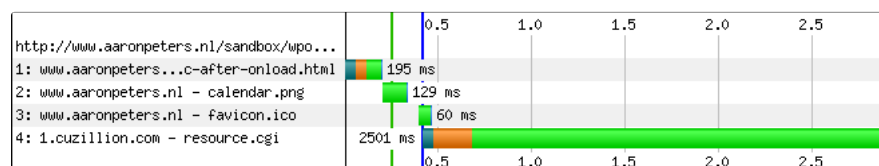
设想你想要加载一些内容在一个iframe，但对于页面不是很重要的，或者不一定要立刻显示iframe内容给用户看，因为隐藏在链接或tab之后下载，可以考虑延迟加载直到主页面加载完后。

```
<script> //doesn't block the load event function createIframe(){ var i = document.createElement("iframe"); i.src = "path/to/file"; i.scrolling = "auto"; i.frameborder = "0"; i.width = "200px"; i.height = "100px"; document.getElementById("div-that-holds-the-iframe").appendChild(i); } // Check for browser support of event handling capability if (window.addEventListener) window.addEventListener("load", createIframe, false); else if (window.attachEvent) window.attachEvent("onload", createIframe); else window.onload = createIframe; </script>
```

The Iframe After Onload technique will consistently show the following behaviour in all browsers:

iFrame在onload之后技术在所有的浏览器中始图显示下面的行为

我做了一些测试在Iframe After Onload test page页面上，并用webpagetest.org验证了，正如期望一样，onload没有被阻塞



Iframe After Onload - Performance - IE8 waterfall chart