**[翻译]CSS选择器的性能影响**

* 原文：[Performance Impact of CSS Selectors](http://www.stevesouders.com/blog/2009/03/10/performance-impact-of-css-selectors/)
* 原作者：[Steve Souders](http://stevesouders.com/bio.php)
* 翻译：[ytzong](http://www.99css.com)

一些关于CSS选择器性能的讨论引起了我的兴趣。

第一个是Shaun Inman写的《[合格的CSS选择器（CSS Qualified Selectors）](http://www.shauninman.com/archive/2008/05/05/css_qualified_selectors)》，实际上这篇博文并没有提到CSS性能，不过有一个来自[David Hyatt](http://webkit.org/blog/)(Safari 和 WebKit 的架构师,同时为 Mozilla, Camino, Firefox 工作)的评论：

如果你非常在意页面的性能那千万别使用CSS3选择器。实际上，在所有浏览器中，用 class 和 id 来渲染，比那些使用同胞，后代选择器，子选择器（sibling,  
descendant and child selectors）对页面性能的改善更值得关注。

接下来的一篇博文很神奇。Jon Sykes做了一个系列的文章来测试CSS性能：[第一部分](http://blog.archive.jpsykes.com/151/testing-css-performance/),[第二部分](http://blog.archive.jpsykes.com/152/testing-css-performance-pt-2/),[第三部分](http://blog.archive.jpsykes.com/153/more-css-performance-testing-pt-3/)，其中[第三部分](http://blog.archive.jpsykes.com/153/more-css-performance-testing-pt-3/)非常值得一读。这使我确信优化CSS选择器是页面性能优化的一个关键步骤。

但是，有两件事一直困扰着我。首先，大量的DOM元素和CSS规则。页面包含60000个DOM元素和20000条CSS规则。这使浏览器非正常的运行（下面将证明这点）。下面的统计表格为美国10大网站的比较：

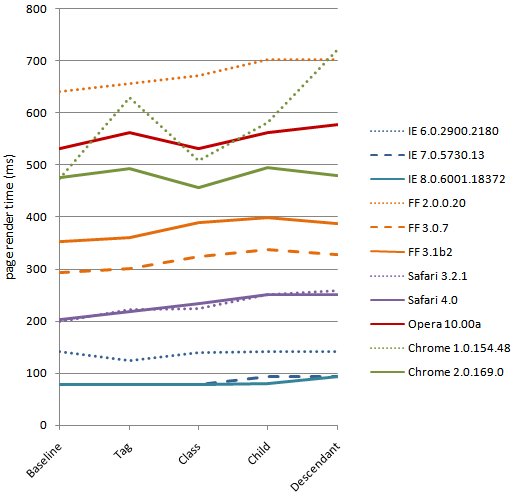
|  |  |  |
| --- | --- | --- |
| **网站名称** | **CSS规则** | **DOM元素** |
| AOL | 2289 | 1628 |
| eBay | 305 | 588 |
| Facebook | 2882 | 1966 |
| Google | 92 | 552 |
| Live Search | 376 | 449 |
| MSN | 1038 | 886 |
| MySpace | 932 | 444 |
| Wikipedia | 795 | 1333 |
| Yahoo! | 800 | 564 |
| YouTube | 821 | 817 |
| 平均值 | **1033** | **923** |

第二件困扰我的事情是测试以多大页面为基准？我想解决的问题是：无效率的CSS选择器会拖慢页面吗？所有测试的5个页面包含20000个a元素（嵌套在P, DIV,DIV, DIV内部）（译注：使用同样的HTML页面），不同的是CSS：基准（没有CSS）（译注：下图中的baseline），标签选择器（tag selector )（a标签有一条CSS规则）（译注：下图中的tag），20000个class选择器（class selectors）（译注：下图中的class），20000个子选择器（child selectors）（译注：下图中的child），20000个后代选择器（descendant selectors）（译注：下图中的descendant）。后面3个页面的大小都超过了3M，而基准及标签选择器的页面则只有1.8M。

接下来调整为：

* 2000个a标签和2000条CSS规则（原先是20000），这实际上有6000个左右的DOM元素，因为a嵌套在P, DIV, DIV, DIV内部
* 基准页面和标签选择器页面像其他页面一样有2000条CSS规则，不过只有简单的class规则生效，不会影响其他含有class属性的标签

我在12个浏览器上进行了测试 。页面呈现时间是用嵌在头部和底部的脚本来测量（所有测试在本地运行）。测试结果如下图（我没有测试Oprea 9.63，你可以下载[csv格式的数据](http://stevesouders.com/tests/css-selectors-my-2000-tests-data.csv)，[这里是测试页面](http://stevesouders.com/tests/css-selectors/index.html)）：



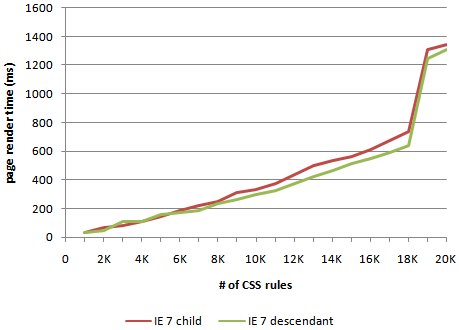
性能随浏览器而改变； 所有测试是在同一台PC机的不同浏览器，不同PC机的不同浏览器可能有不同的性能特点。这个测试的目的不是比较各浏览器的性能，而是为了测试浏览器如何处理逐渐复杂的CSS选择器。

【修订：更深入的比较Firefox 3.0和3.1，我发现这台PC上的Firefox 3.1和IE 8逊于其他PC机。即使再进行测试，也会由于不同PC的硬件差异导致不同的结果，所以，不要用这些数据来比较浏览器。】

毫不意外，越复杂的页面（子选择器和后代选择器）通常性能越差。最大的意外在于第四个页面居然是最差的。所有的浏览器平均慢了50毫秒，观察最大的(IE 6&7,FF3)平均只有20毫秒。对现在70%+的用户来说，改进CSS选择器只能提高20毫秒。

请记住，这些测试接近最坏的情况。20个a标签嵌在P, DIV, DIV, DIV导致总共有6000个DOM元素，这是10大站点中最复杂两个网站（见表一）的两倍大。另外，测试的页面含有2000个极端无效的CSS规则，一般的网站大约有三分之一（译注：网站本身的CSS规则）的子选择器及后代选择器。以Fackbook为例，2882个CSS规则里只有750个极端无效的规则。

为什么我的测试结果和之前其他人的有些不同？一个不同来自于这极其极端。它成倍的放大于我们通常所用的页面。在这种情况下，浏览器面对3M的页面，60000个DOM元素及20000条CSS规则会有不同的表现。我注意到，我的测试结果在IE6&7中非常不同。我想知道IE在处理CSS选择器时是否某个规则。为此我测试了子选择器和后代选择器页面，从1000至20000逐渐增加a标签和CSS规则的数量，结果如下图所示，IE在18000条CSS规则时陡增。但是IE6&7 渲染页面更接近实际，因为在我的测试中，他们的性能竟然是最高的。



基于这些测试我有以下假设：对大多数网站而言，优化CSS选择器活得的性能提升很小，不值得去计较。有些配合Javascript交互的CSS规则会明显的拖慢页面。这是应该关注的焦点。所以我开始关注现实中影响页面性能的CSS样式相关的小问题（offsetWidth,:hover）。

**译注：**

* 这是本人的处女翻
* 并非一字一句的直翻，原则是尽量保留有用的内容
* 水平有限，难免糟蹋了大师的作品，有疑问请自行到原文查阅