



基于 Page Speed 的网站前端性能优化分析*

——以湖北省高校数字图书馆门户为例

聂应高

(咸宁学院图书馆 咸宁 437000)

【摘要】介绍 Page Speed 的产生背景和基本功能,以湖北省高校数字图书馆门户为案例,分析影响网站前端性能的主要因素和存在问题,提出相应改进建议,拟定具体优化策略。

【关键词】性能优化 数字图书馆 门户

【分类号】G250.76

Analysis on Front – end Performance Optimization of Website Based on Page Speed

——A Case Study of University Digital Library Portal in Hubei Province

Nie Yinggao

(Library of Xianning University, Xianning 437000, China)

【Abstract】 This paper introduces the backgrounds and functions of Page Speed. Taking the case of University Digital Library Portal in Hubei Province, it analyses some key factors and problems of website performance, and proposes appropriate recommendations for improvement and formulates specific strategies for optimization.

【Keywords】 Performance optimization Digital library Portal

近几年,网站作为一种强大的工具和平台逐渐融入人们的生活,各类网站每天都在迅速增长。据中国互联网信息中心(CNNIC)《第24次中国互联网络发展状况统计报告》^[1]显示,截至2009年6月30日,中国的网站数,即域名注册者在中国境内的网站数达到306.1万,较2008年末增长6.4%。纵观这些网站性能,有很多网页加载速度慢,有时甚至拒绝访问。Google性能优化专家Steve Souder认为,只有10%–20%的最终用户响应时间花在下载HTML文档上;其余80%–90%时间花在下载页面中的所有组件上^[2],也就是说,80%–90%的用户等待时间是来自于网站前端页面的加载。在现有网络条件和软硬件设备的基础上,如何有效提高网站响应速度是人们最关注的问题之一。开发人员往往都会想到负载均衡、分布式和CDN等性能优化技巧,而对网站前端性能的研究较少。因此,笔者从Google Page Speed^[3]产生背景和基本功能入手,以湖北省高校数字图书馆门户为例,分析影响门户系统前端性能的主要因素,提出相应的改进建议,并拟定具体优化策略,以期为优化网站前端性能提供有益的启示和借鉴。

收稿日期:2009-10-15

* 本文系咸宁学院校级科研资助项目“数字图书馆网页加速与优化分析”(项目编号:KY0832)的研究成果之一。

1 Page Speed 的产生与基本功能

1.1 Page Speed 的产生

Google 的成功很大程度上在于极快的网页响应速度。2009 年 6 月 4 日, Google 开发团队针对 Steve Sounder 网页性能最优方法^[4], 成功地推出一款基于 Firefox/Firebug 的开发类插件——Page Speed, 旨在帮助开发人员分析网站性能存在的主要问题, 并有针对性地提出优化改进意见。它支持的操作系统为 Linux、Mac、Windows XP 或 Windows Vista。在此之前, Google 内部已经广泛使用 Page Speed 优化网页前端性能。

1.2 Page Speed 基本功能

Page Speed 能够用于评估、诊断和优化网页性能, 它与 Firebug 插件无缝结合在一起, 为开发人员带来很大的辅助作用: 方便写出格式化 HTML 代码; 精确编辑 CSS 控制网页模块; 动态调试 JavaScript; 探索 DOM 结构和实时监控网络状态等。Page Speed 可以用来测试分析网站前后端代码, 综合得出一份详细的网页前端性能评估报告, 告诉开发人员如何根据优化建议将网页性能优化到最佳。

Page Speed 主要集成 5 大功能: 全面分析页面性能, 并提供详细的性能优化报告; 内置 JavaScript 及图片优化(含 JS Minify); 改进 Firebug Net 模块的显示; 页面请求活动视图(详细显示各请求加载的时间顺序与时间消耗, 方便找出网页性能瓶颈); JavaScript/CSS 性能优化等。

2 门户性能优化报告与页面活动视图

2.1 Page Speed 性能优化报告

打开湖北省高校数字图书馆门户网站主页(<http://www.hbdlb.cn/>), 待 Firefox 浏览器左下角出现“Done”或“完成”字样, 启用 Page Speed 插件。点击 Page Speed 面板上的“Analyze Performance”按钮, Page Speed 便开始对门户网页性能逐项测试与评估, 并按重要程序和优先级排列, 生成一份详细的页面性能分析报告, 如图 1 所示。

图 1 页面性能分析报告分别用 4 种不同符号描述网页性能: 感叹号(非常重要, 优化后可以非常明显地提高网站响应速度); 三角形(一般重要, 优化后可以比较明显地提高网站响应速度); 对号(基本正常, 不

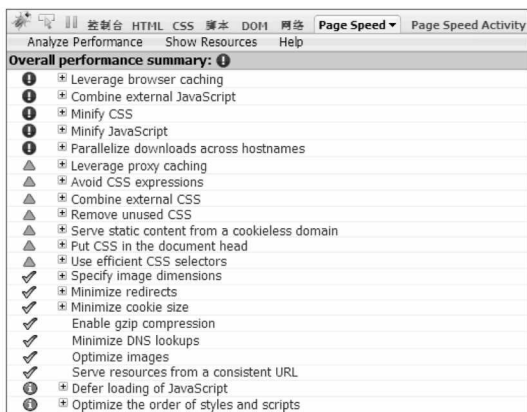


图 1 湖北省高校数字图书馆门户页面性能分析报告

需要特别的优化修改); 消息符(提示信息, 提供额外帮助信息)。展开每条建议前面的“+”可以查看详细描述, 直接点击每条规则就能够查看该规则的具体内容。此外, 点击“Show Resources”, 则能够查看页面中包含的资源及其类型、来源、大小和传输量等信息, 并提供具体优化建议。

2.2 Page Speed Activity 页面活动视图

Page Speed Activity 面板主要用来记录所有浏览器的活动, 展现网页加载各种元素所用的时间(单位: 毫秒)。所有活动记录主要包括 6 个方面: 查看页面耗时间; 查看浏览器事件; 区分页面加载缓慢原因; 获取特定条件 JavaScript 事件列表; 显示多页面加载的加载顺序; 对 Page Speed 优化前后的表现进行对比等。因此, 结合 Page Speed 页面性能分析报告, 开发人员能够进一步对网站前端性能做出评估, 更明确地知道到底是哪些页面元素占用和耗费大量时间, 便于更有针对性地进行优化改进。不同阶段所占用的时间, 记录活动会用不同色块表示。湖北省高校数字图书馆门户页面活动视图如图 2 所示:

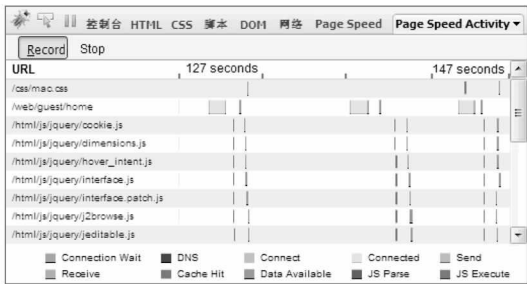


图 2 湖北省高校数字图书馆门户页面活动视图

3 门户前端性能优化分析

3.1 门户建设目标

湖北省高校数字图书馆门户按照中国高等教育文献保障系统(China Academic Library & Information System,CALIS)的标准建设,可分为门户层(IP 统一认证)、服务层(资源检索、资源调度、馆际互借、文献传递、实时咨询等)和资源层(共享资源)三层功能区域,系统界面如图 3 所示。其主要建设目标为:依托现代网络条件和先进的数字图书馆技术,以各高校图书馆丰富的文献资源为基础,建立纸本馆藏文献与数字文献相结合的文献资源体系和网上公共服务体系,允许合法高校图书馆将湖北省高校数字图书馆门户提供的各个应用、资源嵌入各馆网站,进一步丰富各馆馆藏资源与知识服务,实现信息资源共建、共知与共享,为湖北省各高校教学、科研和重点学科建设提供高效率的文献服务。



图 3 湖北省高校数字图书馆门户系统功能界面

3.2 门户性能优化建议

基于 Google 的 21 项优化标准,笔者利用 Page Speed 对湖北省高校数字图书馆门户性能进行大量测试研究,从优化级别、优化得分和主要优势(影响)等方面详细分析门户前端性能存在的主要问题,并给出相应的改进建议,如表 1 所示:

表 1 湖北省高校数字图书馆门户前端性能优化分析

优化选项(标准)	优化级别	得分(%)	主要优势(影响)	优化改进建议
利用浏览器缓存	非常重要	49.4	从缓存读取资源,减少与服务器的交互	为 .swf、.gif、.css 和 .js 等比较固定的静态资源指定较长过期时间,用户通过本地缓存可以更快地访问网站
合并外部 JavaScript	非常重要	0	减少 HTTP 请求数,加快页面读取速度	适当对门户中 43 个 JavaScript 文件进行合并,要尽可能减少 JavaScript 文件数量
精简 CSS	非常重要	40.8	符合 Web 标准开发,文件体积小,加载快	把具有相同属性的元素合并。门户中有用 CSS 大小为 77.7KB,通过精简可以节省 23 KB(约占 29.6%)
精简 JavaScript	非常重要	32.0	符合 Web 标准开发,文件体积小,加载快	尽量压缩 JavaScript,门户中 JavaScript 总大小为 565.6KB,精简后可以节省 192.4KB(约占 34.0%)
跨主机名并行下载	非常重要	0	使用多个不同主机名来增加并行下载	门户主页中含有 47 个并行请求,可以通过跨多主机名(4-5 个)分配这些请求,增加并行下载数量
利用代理缓存	一般重要	67.2	代理缓存对网页整体性能有显著提升	使用"Cache-Control: Private" or "Vary: Accept-Encoding" 控制网页缓存,例如门户中公共缓存、可压缩 JavaScript 和 CSS
避免使用 CSS 表达式	一般重要	56.0	代价极高,有时甚至会造成页面假死	门户中 4 个 CSS 表达式要尽量避免使用。CSS 表达式似乎仅支持 IE 浏览器,计算频率太高,执行效率极低
合并使用外部 CSS	一般重要	0	减少 HTTP 请求数,加快页面显示速度	建议尽量将门户中 13 个 CSS 文件适当合并,以减少 CSS 文件的数量
删除未使用的 CSS	一般重要	54.4	减少信息的发送量,加速页面反应速度	门户中 CSS 总大小为 77.7KB,没有利用的 CSS 大小为 38.9 KB(约占 50.1%),尽量删除这些文件
在无 Cookie 的域处理静态内容	一般重要	45.4	可以有效减少页面请求的总大小	尽量在无 Cookie 的域处理静态资源(CSS,JavaScript 及 GIF 等),门户中要为这些资源发送 4.8 KB Cookie 请求
CSS 放在页面顶部	一般重要	0	便于网页有序加载,能够更快渲染页面	建议使用 Link 将 Body 块中的 mac.css 文件放置于 Head 区域中,加快页面渲染效果
使用有效率的 CSS 选择符	一般重要	0	高效率 CSS 选择符可以加快页面呈现	优化 Base、Navigation、Portlet、Tabs、Green、Orange、Hubei 和 Mac 等样式文件的 CSS 选择符
指定图片尺寸	基本正常	95.0	避免计算图片尺寸和重定位页面内容	需要为门户中所有图片的高和宽设置一个特定值,以便加速网页显示
减少重定向	基本正常	82.0	每增加一次重定向就增加一次 Web 请求	URL 链接末尾需要增加"/",避免无意义的跳转,尽量删除门户中的 3 个重定向
减小 Cookie 大小	基本正常	100.0	减少用户响应时间,能很好地提高效率	尽量减小 Cookie,删除无用的 Cookie,合理设置过期时间。主页中所有请求的 Cookie 平均大小为 89KB
启用 Gzip 压缩	基本正常	100.0	减少网络传输时间,加快网页下载速度	启用 Gzip 压缩静态内容可减少带宽支出,大概节省 70% 的传输空间,可使用户响应时间缩减 50% 左右
最小化 DNS 查询	基本正常	100.0	减少 DNS 查询可以加快页面加载速度	DNS 查询的开销很大,过多使用站外 Widget 很容易引起过多 DNS 查询。每个页面一般只含 2-4 个域名
优化图片	基本正常	100.0	优化图片有利于节省更多流量与空间	增加搜索引擎可见的文本描述,以及在保持图像质量的情况下尽量压缩图像的文件大小
始终从一个 URL 提供服务	基本正常	100.0	相同资源使用相同的 URL,便于缓存	从唯一的 URL 提供资源服务,消除重复下载字节与额外 RTTs
推迟加载 JavaScript	提示信息	禁用	推迟加载 JavaScript,以免阻塞其他资源下载	推迟加载 JavaScript 函数调用,优先执行其他功能代码。但在 Firefox 中被禁用,因为 Firefox 不支持 Defer 属性
优化样式和脚本的顺序	提示信息	n/a	CSS 放于 JavaScript 上面,加快用户感知速度	按正确顺序加载网页风格与内外脚本,能够更好地加快并行下载与页面呈现速度

3.3 影响门户性能的主要因素

湖北省高校数字图书馆门户系统采用模块化设计方式,但从表1各项数据分析,该门户系统有19项涉及到Google标准(推迟加载JavaScript被禁用,由于Firefox不支持Defer属性),其中前12项(约占63.2%)对前端性能有较大负面影响,如表2所示:

表2 影响湖北省高校数字图书馆门户前端性能的主要因素

优化级别	使用到的优化标准总数	所占比例	优化原因及效果
非常重要	5项	26.3%	这5项严重影响门户前端性能。检修JavaScript与利用浏览器的缓存特性,可以极大降低网页对服务器性能的消耗,达到与本地访问相当的速度,非常明显地提高网站响应速度
一般重要	7项	36.9%	这7项对门户前端性能有一定负面影响。优化改进门户中6项CSS(其中3项得分为0),可以有效加快网页渲染效果,比较明显地提升网页整体性能
基本正常	7项	36.9%	5项得分为100%,另两项接近100%。这些标准处理得非常恰当,运行良好,不需要作其他特别的性能优化,基本上没有造成额外开销

总体来说,该门户系统前端性能存在一定缺陷,主要有以下三个方面因素:

(1)缓存。门户中有两项标准都涉及到缓存:使用浏览器缓存(分析报告中位于首位,得分仅为49.4%)、使用代理缓存(分析报告中位于第6位,得分为67.2%)。这两项都严重影响门户前端性能,需要进一步优化改进。

(2)JavaScript。Google专家认为,如果网站前端JavaScript不做详细检修优化,性能改善就不会明显。从性能优化分析报告可以看出,门户中存在大量需要优化的JavaScript,两项JavaScript标准优化级别均显示“非常重要”,得分为0和32.0%。可见,这两项标准都严重地影响前端性能,需要重点优化改进。

(3)CSS。门户中的CSS(包括CSS大小、位置、数量和选择符等)对浏览器的渲染有很大负面影响,不利于网页被搜索引擎收录。比如,分析报告中6项CSS标准得分都很低(其中3项得分为0),需要进一步优化改进。

4 门户前端性能优化策略

针对Page Speed性能改进建议,结合影响湖北省高校数字图书馆门户性能的主要因素,笔者从以下几个方面分别探讨相应的性能优化策略。

4.1 善于利用缓存

浏览器(代理)使用缓存来有效减少HTTP请求次数和响应数据量,以达到更快的网页载入速度。通过设置Expires Header属性,当用户再次访问页面时,直接从缓存中读取文件,不需要重新下载。对于静态资源(图片,JavaScript,CSS,Flash和ICO等不需要经常改动的文件)可以使用Expires Header来缓存;对于动态资源,通过Cache-Control文件头来帮助浏览器实现有条件的请求。

比如用文本编辑器创建一个.htaccess文件,代码如下:

```
<FilesMatch "\.(ico|js|css|swf|pdf|flv|jpg|jpeg|png|gif)" $ "
> Header set Expires "Thu, 31 Dec 2009 20:00:00 GMT"
</FilesMatch>
```

将.htaccess文件上传到门户根目录,就能够轻松将静态资源缓存时间设为2009年12月31日(也可以设置为将来任意某个日期)。

需要注意的是,Expires Header使用的是一个特定缓存时间,要求服务器和客户端有严格的时钟同步,而且一旦缓存过期时间到来,新的过期时间需要重新配置。另外,开发人员更新静态资源以后,还必须重新命名已经更新的文件,否则使用缓存的用户将无法看到更新。

4.2 优化JavaScript/CSS^[5]

JavaScript/CSS优化对于提高门户前端性能的作用非常大。门户系统的JavaScript/CSS优化改进主要从以下几个方面进行:

(1)合并JavaScript/CSS。尽可能将多个JavaScript/CSS写入一个.js/.css文件,能够减少浏览器HTTP请求次数,加快用户下载和呈现速度,减轻Web服务器负载,而且对于提高用户体验和蜘蛛爬行速度都是有益的。可见,最理想的情况是将所有JavaScript/CSS合并成一个.js/.css文件。

(2)将CSS置于页面顶部。建议使用Link将CSS放置于Head区域,因为页面一旦被加载,Link引用的CSS会同时被加载,加快页面渲染效果。而使用@import引用CSS,只有等到页面全部下载完成再加载。

(3)将脚本置于页面底部。尽量将脚本置于页面底部,浏览器解析脚本需要调用多个组件,一旦遇到<script>标签,就会先执行脚本任务,完成后才会启动其他下载。因此,将脚本放于底部不仅能加快网页

载入速度,还能有效回避蜘蛛对 JavaScript 代码的分析。另外,浏览器每次遇到 <script> 标签就会启动 JavaScript 分析引擎,开销很大。

(4) 使用外部 JavaScript/CSS。在实际应用中,网页使用外部文件普遍会快些。因为内置在 HTML 中的 JavaScript/CSS 会在每次请求中随 HTML 文档重新下载,不利于浏览器缓存。更重要的是,使用外部 JavaScript/CSS 能够使网站更新维护更为高效方便。如脚本或样式表需要修改时,只需修改一个 .js/.css 文件,所有引用该 .js/.css 文件的页面会自动生效。

(5) 精简 JavaScript/CSS 代码。精简代码的主要工作是去除多余的和不必要的字符(注释、空格、换行、Tab 缩进等),以减小文件的大小。开发人员可以直接将 Page Speed 精简后的 JavaScript/CSS 文件上传至服务器。这是最简单、最有效的解决方法,不需要对 HTML 页面进行任何修改,比较容易实现。另外,JSMIn 也是目前使用最广泛的 JavaScript/CSS 精简工具。

(6) 删除重复脚本。尽量避免重复脚本,因为重复脚本能够导致浏览器分析引擎启动两次,有时甚至请求文件两次,而且当使用第三方 JavaScript 库时,重复脚本可能会导致问题的产生。在 Internet Explorer 中重复引用脚本会产生不必要的 HTTP 请求。

(7) 避免使用 CSS 表达式。表达式计算频率太高,执行效率极低,不能通用,比如,IE8 的标准模式就不再支持。同时,CSS 表达式暴露了一个脚本执行环境,构成一个可能的脚本注入攻击方向。另外,表达式支持更多特定的功能,基本上可以通过标准的 JavaScript

来实现,通常是通用的而且能够以较低的成本运行。

5 结 语

本研究虽然只是着眼于湖北省高校数字图书馆门户前端性能,但是由于基于 Page Speed 性能优化分析报告,因此提出的优化思路、改进建议和基本策略具有一定的普遍适用性,可以应用于不同类型的网站。好的工具对于提高工作效率的确有很大的帮助,但好的工具只是一方面,更重要的是进一步学习网站优化知识,多与同行专家学者交流沟通,这样开发人员才能够针对各网站特色和实际情况,做出最切实际、最有效的优化策略,并付诸实施。

参考文献:

- [1] 中国互联网络信息中心. 第 24 次中国互联网络发展状况统计报告[EB/OL]. [2009-07-18]. <http://www.cnnic.net.cn/uploadfiles/doc/2009/7/16/125040.doc>.
- [2] 桑德斯. 高性能网站建设指南[M]. 刘彦博译. 北京:电子工业出版社,2008.
- [3] Using Page Speed[EB/OL]. [2009-09-20]. <http://code.google.com/speed/page-speed/>.
- [4] Web Performance Best Practices[EB/OL]. [2009-09-20]. http://code.google.com/intl/zh-cn/speed/page-speed/docs/rules_intro.html.
- [5] Steve Solders. High Performance Web Sites[M]. Sebastopol: O'Reilly, 2007.

(作者 E-mail: libnyg@163.com)