

3G 院线业务应用层的性能优化^①

张会云, 王 纯

(北京邮电大学 网络与交换技术国家重点实验室, 北京 100876)
(东信北邮信息技术有限公司, 北京 100191)

摘 要: 提出了对 Web 性能前端进行改进的方法。3G 院线网站开发完成以后, 无论是通过管理员的使用, 还是 Yslow 工具的分析, 都表明该网站性能较低。整个网站加载速度慢, 加载文件多。针对这一缺点, 对 3G 院线网站进行了改进。主要是减少 HTTP 请求数、对文件进行压缩传输和在浏览器中对文件进行缓存来达到提高网站访问性能。

关键词: 性能优化; WEB; 网站; 3G 院线

Performance Improvement of Service Application Layer in 3G Movie Platform

ZHANG Hui-Yun, WANG Chun

(State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)
(EBUPT Information Technology Co. Ltd., Beijing 100191, China)

Abstract: This paper promotes ways to improve the performance of website. The management website of 3G Movie platform doesn't work very well either when analyzed by tools like Yslow or when being used by operator after it has been deployed. It seems to work slowly because there are too many files to load. For the shortage, this paper puts forwards four ways to improve the process and performance of the website, such as, making fewer HTTP requests, compressing components, caching files in browser and so on.

Keywords: performance improvement; Web; website; 3G Movie platform

1 引言

3G 院线业务应用层主要是为 3G 院线系统提供管理平台, 即 3G 院线管理网站。该网站采用 ExtJS 和 FleaPHP 框架实现, 为管理员提供了简洁美观的页面呈现效果。该网站主要管理系统中五种数据(影片数据、EPG(Electronic Program Guide, 电子节目菜单)数据、升级包数据、上线时间数据和套票数据)的生成、审批以及将审批后的数据下发至 MBMS(Multimedia Broadcast Multicast Service, 多媒体广播多播业务)平台和终端等过程。MBMS 平台和终端对各种数据的接受和处理情况都需要通过该网站才能及时地展现

出来, 同时也只有这样, 管理员才能知道影片及其相关数据是否已被正确地下载到终端。所以维护这个网站的重要性不言而喻。

2 存在的问题

3G 院线管理网站自开发完成并投入使用以来一直在比较稳定地运行, 而且页面设计方式、呈现效果也得到了管理员的认可。然而该网站虽然从功能上完成了 3G 院线系统所有的业务需求, 但自投入使用以后, 管理员提出了如下三个问题: ①网站登录时间长; ②部件加载时间长; ③页面刷新慢。

^① 基金项目: 国家杰出青年科学基金(60525110); 国家 973 计划(2007CB307100, 2007CB307103); 国家自然科学基金(60902051); 中央高校基本科研业务费专项资金(BUPT2009RC0505); 电子信息产业发展基金

收稿时间: 2010-07-17; 收到修改稿时间: 2010-08-15

根据上述问题,对该网站进行了性能测试,根据反馈数据进行总结,形成表格,如表 1 所示:

表 1 改进前 3G 院线网站性能表

场景	加载文件数 doc/js/css/cssimage/image	文件大小/ 传输大小/ 压缩率	加载 时长	说明
登录 页面	28 (1/4/1/19/3)	1.141M/ 1.141M/ 0%	2.354s	登录时长
主 页 面	133 (1/114/2/14/2)	1.881M/ 1.881M/ 0%	4.267s	部件加载时 长
刷 新 主 页 面	133 (1/114/2/14/2)	1.891M/ 1.891M/ 0%	4.864s	刷新时长

表 1 中提到的问题有可能是系统前后端共同引起的。后端服务器依赖于综合数据业务平台(Integrated Data Platform, IDP),该部分改进方向有多种,比如对服务器中的消息分发模块进行改进^[1],对内存对象的缓存管理方式进行改进^[2]。本文只讨论 web 前端改进方案,对后端服务器的改进方案不再赘述。

为了解 web 前端性能,对 3G 院线管理网站的页面加载性能进行了测试。目前 B/S 结构下页面加载性能测试工具有两个:来自雅虎的 Yslow 和来自 google 的 Page speed。因为雅虎的工具更专业,更标准,应用更广泛,所以这里使用 Yslow。首先简单介绍 Yslow 工具。Yslow 是雅虎公司推出的一款测试性能工具,需要安装在 firefox2.0 版本下,而且安装该工具前还需要安装 firebug 工具。这款软件可以分析出网站的组件数、Apache 的配置参数。通过这些参数的分析,给网站提出相应的改进意见。分析测试结果分为 A-F 级别,其中 A 为最好, F 为最差。

表 2 是 3G 院线网站的 Yslow 测试情况:

表 2 3G 院线性能测试结果表

具体测试 页面	测试结 果	测试结果页面加载各项性 能参数指标	参数指标测试 结果
主 页 面	F	Make fewer HTTP requests	F
		Add Expires headers	F
		Compress components with gzip	F
		Put CSS at top	F
		Put JavaScript at bottom	F

Yslow 工具在对网站进行分析的时候有 13 项参数指标。但是对 3G 院线网站进行 Yslow 分析以后,发现只有表 2 中提到的几项指标使得 3G 院线网站的总体测试结果为 F。所以本文针对这几项给出相关的解决方案,具体如下。

3 解决方案

下面我们通过一个表格来看一下为什么 Yslow 会分析出 3G 院线网站在表 2 中提到的 5 项指标中测试结果均为 F:

表 3 3G 院线前端性能分析表

性能参数指标	原因定位	改进建议
Make fewer HTTP requests	113 个 JavaScript 脚本	把这些 JS 文件尽量合并成一个
	15 个背景图片	使用 CSS sprite 合成图片文件
Add Expires headers	128 静态组件没有设置过期时间	给这些文件增加 expire 头
Compress components with gzip	116 个文本文件没有进行压缩传输	采用压缩文件的方式对文件进行传输
Put CSS at top	一个样式表在文档 body 部分	把 CSS 样式表文件放在文档的头部
Put JavaScript at bottom	113 个 JavaScript 脚本文件在文档 head 部分	把 JS 文件放在文档的 body 部分

从表 3 中不难看出,前端性能低下的原因就是表 3 提到的 6 种原因。究其根本原因主要是脚本数、CSS 样式表数、零散图片数多,所以提出解决方案如下:

- 1) 减少 HTTP 请求数。
- 2) 缓存组件。
- 3) 压缩组件。
- 4) 把 CSS 样式表放在顶部,脚本文件放在底部。

4 设计实现

4.1 减少 HTTP 请求数

性能黄金法则(Performance Golden Rule)指出只有 10%-20%的最终用户响应时间是花在接受所请求的 HTML 文档上。剩余的 80%-90%的时间花在 HTML 文档所引用的所有组件(图片、脚本、样式表、Flash 等)进行 HTTP 请求上^[3]。同

时从表 3 中不难看出 HTTP 请求之所以数量多,是因为该页需加载 113 个脚本文件请求和 15 个图片请求。

一种可选的实现方式是直接从页面中去移除组件。这种实现方式确实可以减少 HTTP 请求数,提高网站访问速度,但组件的删除会影响界面的呈现效果。尤其是脚本文件的移除,会导致性能和产品设计之间的矛盾。下面介绍一下在不改变用户体验的基础上,减少 HTTP 请求数的方法。

4.1.1 减少脚本文件 HTTP 请求数

首先分析 3G 院线网站的脚本文件的组织方式。了解到原脚本文件是以功能(比如增加、删除、查找等)为单位进行组织的,而不是以模块为单位进行组织。考虑到函数名冲突的可能性,先将某模块下的所有脚本文件进行合并,合并时保证各脚本文件中变量不冲突;再将所有的模块脚本文件再合并。这样手动编写的脚本文件被合并成一个。

对脚本文件进行合并的实现方式有多种,比如 PHP 脚本的方式, shell 脚本的方式。鉴于论文篇幅的原因,只介绍简短的 shell 脚本方式:

```
#!/bin/bash
if [ $# == 1 ]
then
    `rm test.js`
    for file in `find $1 -name '*.js'`
    do
        echo $file
        echo "/* $file */" >> test.js
        `cat $file >> test.js`
    done
fi
```

最后把 113 的脚本文件合并成了 1 个手写 JS 文件、2 个 ExtJS 自带 JS 文件和 1 个 Fusioncharts 插件自带 JS 文件。

4.1.2 减少图片文件 HTTP 请求数

在不减少图片的情况下,减少图片文件 HTTP 请求数的主要思路是将多张图片进行合并,通过对新图片定位来获取原图片。合并图片的方式有两种:图片地图(Image Maps)和 CSS Sprites。和图片地图相比,CSS Sprites 更具有灵活性。例如图片地图要求图片必须是连续的,但是 CSS Sprites 却没有这个限制。同时

CSS Sprites 还减低了下载量,它合并后的图片会比分离的图片之和小。所以这里只介绍 CSS sprites 的方法,如下:

首先选择合适的图片合并工具。目前网上实现 CSS Sprite 的工具比较多,比如 bg2css、CssBgImageMergeTool 等,本文采用后者。再将 Yslow 分析出的零散图片文件通过该工具进行合并。因为该工具还生成相应的 CSS 样式表,只需要把相应的样式合并到系统 CSS 样式表中即可生效。这样多张图片的 HTTP 请求就减少为 1 张图片的 HTTP 请求。

最后把 15 张图片合并成了 1 张 3G 院线图片文件和 10 张 ExtJS 框架自带的图片。

4.2 缓存组件

在 4.1 节中提到需要减少 HTTP 请求数,即尽量减少组件的个数。但是组件个数的减少也即意味着组件大小的增加(比如合并后的 JavaScript 脚本文件)。如果每次登录页面都需要向服务器请求相应的组件,则网站的平均响应时间也会增加。考虑到脚本文件稳定了以后都不太会修改,这里提出缓存组件的概念。具体实现方式有两种。

4.2.1 添加 Expires 头

该操作主要作用是通过配置服务器告诉浏览器对相关组件进行存储直到未来的某个时刻,或者是某段时间以后。3G 院线配置方式如下:在 Apache 中动态安装 mod_expires 模块,安装完成以后,在 httpd.conf 文件增加命令行如下:

```
<IfModule mod_expires.c>
    ExpiresActive On
    ExpiresByType image/gif A2592000
    ExpiresByType image/jpeg A2592000
    ExpiresByType image/png A2592000
    <FilesMatch "\.(php|js|css)$">
        ExpiresDefault A2592000
    </FilesMatch>
</IfModule>
```

其中各参数意义如下:

ExpiresActive On: 表示在 HTTP 消息中添加 Expires 头

ExpiresByType 文件类型 时间长: 表示对该类型文件添加 Expires 头。时间长参数指定了该文件的过期

时间,其中 A 表示时间起始点是该组件被客户端访问的时间。

FilesMatch 文件名正则表达式 **时间长**:表示对文件名符合正则表达式的文件添加 Expires 头。时间长参数意义同上。

4.2.2 配置 ETag

ETag 标签和 Expires 头共同使用,用来判断缓存在浏览器的组件是否是最新的组件,是否需要再次请求。3G 院线的配置方式如下:直接在 Apache 的配置文件中 httpd.conf 文件下修改配置如下:

FileETag MTime Size

该参数表明 ETag 标签值是系统时间戳(MTime)+文件大小(SIZE)。

最后通过 Yslow 检测分析到 Add Expires headers 性能指标的结果为 A。

4.3 压缩组件

对组件进行压缩传输主要是减少 HTTP 响应的大小,以便减少网站的响应时间。目前大多数浏览器都支持 GZIP 的方式进行压缩传输。为保证该网站对浏览器的兼容性,也采用 GZIP 方式进行压缩传输。另外考虑到压缩传输需要花费额外的 CPU 周期,为了保证开销和收益的平衡,对大于 1KB 和 2KB 的文件进行压缩传输,而且因为图片格式的文件被压缩了,所以只需对 HTML、脚本文件和样式表文件进行压缩。

3G 管理网站实现方式如下:

3G 院线采用的服务器是 Apache1.3 版本,所以采用 mod_gzip 来配置 gzip。首先是在 apache 下安装 mod_gzip 模块,然后在 httpd.conf 文件中配置参数如下:

```
<IfModule mod_gzip.c>
  mod_gzip_on Yes
  mod_gzip_item_include file ".js$"
  mod_gzip_item_include file ".css$"
  mod_gzip_item_include file ".html$"
  mod_gzip_temp_dir /home/3gfilm/local/apache/tmp
  mod_gzip_minimum_file_size 1000
  mod_gzip_keep_workfiles No
</IfModule>
```

其中各参数意义如下:

mod_gzip_on: 表示是否支持 gzip 压缩;

mod_gzip_item_include file: 表示支持压缩的文件格

式。

mod_gzip_temp_dir: 表示压缩文件存放的路径

mod_gzip_minimum_file_size: 表示最小压缩文件大小。

mod_gzip_keep_workfiles: 表示是否保存压缩文件的暂存文件,这里不保存。

修改 httpd.conf 配置文件后, Yslow 分析得出还有相关的 PHP 动态页面没有进行传送压缩,即使在 httpd.conf 文件配置 PHP 类型文件也进行压缩,还是不起作用。查找相关资料了解到,要对 PHP 页面进行压缩传输,需对每个 PHP 页面增加控制语句如下:

```
<?php
  if
    (substr_count($_SERVER['HTTP_ACCEPT_ENCODING'], 'gzip'))o
    ob_start("ob_gzhandler");
  else
    ob_start();
?>
```

经过上述操作后,再对 3G 院线网站进行 Yslow 检测,分析得出 3G 院线网站所有需要压缩的文件都已经进行了压缩传输。

4.4 样式表和脚本的位置

样式表和脚本的位置之所以重要是因为它们在文档中的位置直接关系到页面内容的加载和呈现效果。

4.4.1 样式表放在顶部

如果把样式表放在页面的底部,那么即使系统已经完全加载了脚本文件,但为了避免出现样式发生变化时重绘页面元素的情况,浏览器会阻止脚本内容的逐步呈现,会导致白屏现象。

所以设计 web 系统的时候需要遵守一个原则就是把样式表放在文档的 HEAD 标签中。

针对这一原则,3G 院线网站将 CSS 样式表放在了 HEAD 标签中。

4.4.2 脚本放在底部

在前面也提到了脚本文件需要放在 CSS 样式表的后面,同时脚本文件最好是放在文档的底部来保证页面逐步呈现和提高下载的并行度。

针对这一原则,3G 院线网站将脚本文件放在了 BODY 标签中。

5 总结

本文针对 3G 院线原管理网站前端性能的问题，通过 Yslow 工具进行分析，定位出性能问题所在。在分析 3G 院线网站的基础上，针对出现的各种问题分别提出了相应的改进步骤。

下面是优化后 3G 院线网站性能表，如表 4 所示：

表 4 改进后 3G 院线网站性能表

场景	加载文件数 doc/js/css/ cssimage/ image	文件大小/ 传输大小/ 压缩率	加载 时长	说明
登录 页面	28 (1/4/1/19/3)	1.138M/ 295.6K/ 约 80%	1.771s	登录时长，因服务器设置 apache Expire 头，多次登录，时间会逐步减少。
主 页 面	21 (1/4/2/13/1)	1.431M/ 314.9K/ 约 80%	2.79s	部件加载时长
刷 新 主 页 面	22 (1/4/2/13/2)	1.432M/ 315.2K/ 约 80%	2.396s	刷新时长。文件在本机有存储，所以该时间和部件加载时间相比更小。

从表 4 和表 1 的对比中不难看出，通过上面 4 步改进。无论是加载文件数，还是文件传输的压缩率，和在本机的存储的文件量都有大大的提高，这也很好地证明了改进后 3G 院线网站性能有了很大的提高。

另外表 2 中提到的各种性能指标，目前测试结果都是 A 了。这也进一步说明 3G 院线网站性能确实有了很大的提高。

当然 3G 院线网站目前应用还不是很广泛，如果是大范围内应用的 web 服务器，比如 google、百度、淘宝等大型网站，这些大型 web 网站的性能提高还需要参考文献^[4]。

参考文献

- 1 杨妙,王晶.IDP 中消息分发模块的改进.电信工程技术与标准化,2009,(6):73-76.
- 2 王文林,廖建新,朱晓民.VoiceXML 语音平台缓存技术综述.通信学报,2007,28(2):101-108.
- 3 Souders S.刘彦博译.高性能网站建设指南.北京:电子工业出版社,2008.
- 4 曹恺,王海芹,杨永侠.提高高访问量交互式网站性能.计算机工程与应用,2007,(12):170-172.

(上接第 40 页)

用户名与密码。

在个人信息管理模块中，注册用户可以修改自己的信息，尤其是密码信息。这是因为信息安全要求人们经常变更自己的密码。具体实现是要求用户输入新的密码，然后用新的密码替换用户信息表中的旧密码。

修改密码界面^[3]：

用户名:

请输入要修改的密码:

请输入新密码:

图 5 修改密码界面图

5 结论

本文用 UML 的方法对项目需求、关键领域类、需求的内部活动图、系统数据模型进行了分析与阐述。完成系统的数据库表设计，并从面向对象的角度设计了对于表的存取接口函数。

通过 UML 方法，成功的指导了 BBS 系统的系统设计工作，因为 UML 建模不但可用于系统设计的不同阶段，且不同的模型可从不同的视角，以不同的详略程度对系统进行描述。

参考文献

- 1 黄国瑜.数据结构.北京:清华大学出版社,2002.
- 2 王珊.数据库系统简明教程.北京:高等教育出版社,2005.
- 3 明日科技.ASP 数据库系统开发.北京:人民邮电出版社,2006.