

文章编号: 1005-8451 (2009) 02-0035-03

## 播客和视频广告技术的研究及应用

沈 哲, 揭金良

(成都理工大学 信息工程学院, 成都 610059)

**摘 要:** 主要介绍播客的开发原理以及实现方法, 讨论了如何从技术上更进一步优化播客系统的方法。着重阐述如何使用各种解码器对视频文件进行转换, 以开源的解码器 ffmpeg 和 mencoder 为开发工具, 它们可根据用户的特殊需求修改解码器源码, 提高了播客系统的有效性。

**关键词:** J2EE; 播客; 视频广告; ffmpeg

**中图分类号:** TP39

**文献标识码:** A

### Research and application on podcast and video advertisement

SHEN Zhe, JIE Jin-liang

(College of Information Engineering, Chengdu University of Technology, Chengdu 610059, China)

**Abstract:** It was presented the principle of development and the method of implementation for podcast discusses how to further optimize the Podcast System. the develop focused on how to use a variety of video decoder to convert, by open-source decoder ffmpeg and mencoder for development tools, which could be amending decoder source code based on the user's special needs and improve the effectiveness of the system podcasts.

**Key words:** J2EE; podcast; video advertisement; ffmpeg

随着 Web2.0 时代的到来, 传统的传媒方式跟着也发生了翻天覆地的变化, 产生一种新的传媒方式播客, 它具有将自主的信息作为传播方式, 以音视频为主的信息作为传播内容, 与其他传媒形态进行交融的特性。随着播客的普及, 其商业价值也逐渐体现出来, 视频广告就是播客技术结合商业的产物。

## 1 播客

### 1.1 播客介绍

播客, 即 Podcasting, 这一概念最早出现在 2004 年 2 月 12 日英国《卫报》的一篇题为《听觉革命: 在线广播遍地开花》的文章中。2004 年 8 月 13 日, IPOD 的发明者美国人亚当·科利开通了世界第一个播客网站——“每日源代码”(www.dailysourcecode.com), 亚当·科利也因此被称为“播客之父”。

时至今日, 在中国已经涌现出一大批播客网站, 并且得到迅速发展, 并且在 2008 年 2 月由国家颁布了《互联网视听节目服务管理规定》, 该规定

的颁布使得一大批国内的播客网站可以继续发展。

### 1.2 播客原理

一般播客主要采用如下的步骤进行开发:

(1) 网页文件调用上传组件进行视频文件的上传; (2) 后台程序调用 ffmpeg 解码器对上传的视频进行压缩, 输出成 flv 格式的视频文件; (3) 使用 flvtools 处理 flv 文件, 标记上时长、帧速、关键帧等元数据, 这样的 flash 文件才可以拖放; (4) 使用 ffmpeg 产生 flv 文件的缩略; (5) 使用适当的 flv 播放器在网页中调用播放服务器端生成的 flv 视频。

### 1.3 ffmpeg、mencoder 介绍

ffmpeg 是一个开源免费的项目, 它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库 libavcodec, 为了保证质量和性能, libavcodec 里很多 codec 都是从头开发的。FFmpeg 在 Linux 平台下开发, 但它同样也可以在其它操作系统环境中编译运行, 包括 Windows 等。国内比较流行的影音播放器暴风影音采用的就是 ffmpeg 解码器。

mencoder 是一个 Linux 下的视频播放工具 (MPlayer) 自带的编码工具, 它支持各种视频格式, 不过使用起来需要用命令行手工添加所需的

收稿日期: 2008-08-01

作者简介: 沈 哲, 在读硕士研究生; 揭金良, 教授。

各项参数。

ffmpeg 与 mencoder 各有优势, ffmpeg 转换速度比 mencoder 快, 不过 mencoder 支持的格式比 ffmpeg 多。因此, 结合两个解码器的优点同时使用它们, 将大多数的格式交给 ffmpeg 处理, 将 wmv9、rm、rmvb、rt 等 ffmpeg 无法处理的格式采用 mencoder 将其转换成 avi 格式, 再通过 ffmpeg 转换成 flv 格式。

2 基于 J2EE 的播客、视频广告应用架构设计及实现

2.1 基于 J2EE 的播客、视频广告架构介绍

播客、视频广告应用于某大型产业信息网, 该系统采用基于 B/S 的架构。普通注册用户可以通过上传视频文件形式发布播客, 企业注册用户可以发布视频广告以及参与广告的竞价。视频广告的形式类似于播客, 因此代码完全可以复用。

为了方便实现模块化开发、提高代码的复用性和维护性, 本系统采用 J2EE 的架构进行开发。本系统分为 4 层分别为: 数据库、持久层、业务逻辑层和表现层。持久层采用 hibernate 框架的 O/R Mapping 映射技术, 实现面向对象的数据存储。业务逻辑层实现整个系统的业务逻辑。表现层又分为视图层和控制器层, 其中视图层部分采用 Ajax 进行数据传输, 控制器层采用 struts 框架。整个系统通过 spring 框架作为核心框架整合其余的 hibernate 和 struts。其中视频文件的上传和视频格式的转换, 采用编写单独的工具类的形式来对解码器进行操作, 方便代码的复用。显示本播客系统的视频处理流程如图 1。

2.2 使用 ffmpeg、mencoder 解码器

为了对处理解码器的代码进行复用, 将该部分的代码编写了一个工具类。由于视频格式的转换有 2 种方式, 因此, 此工具类需要至少 2 个函数方法, processFLV() 方法用于处理 ffmpeg 解码器将视频文件转换成 flv 格式, processAVI() 方法用于处理 ffmpeg 解码器转换不了的格式, 先用 mencoder 解码器转换成 avi 的格式, 然后主函数再调用 processFLV() 方法转换成 flv 格式。ffmpeg 和 mencoder 的使用很大程度上需要运用各种参数, 因此这 2 个方法均采用 List 的形式将解码器的

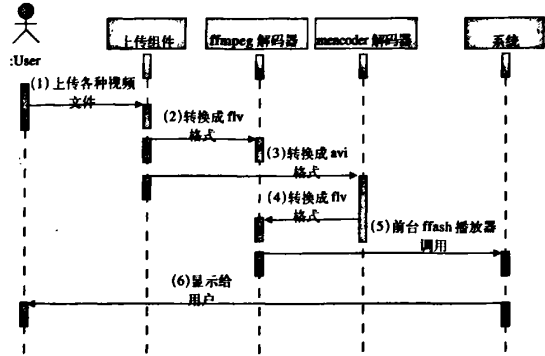


图 1 播客系统的视频处理时序图

各种参数放入 List 集合中。下面展示的是处理 FLV 格式的 processFLV() 方法的代码片段:

```
List<String> commend = new java.util.
ArrayList<String>();
commend.add(softFLVPath);
commend.add("-i");
commend.add(this.oldFilePath);
commend.add("-ab");
commend.add("64");
commend.add("-qscale");// 视频加清晰
commend.add("6");// 4 更清晰
commend.add("-acodec");
commend.add("mp3");
commend.add("-ac");
commend.add("2");
commend.add("-ar");
commend.add("22050");
commend.add("-b");
commend.add("230");
commend.add("-r");
commend.add("24");
commend.add("-y");
commend.add(this.newFilePathFLV);
```

对 ffmpeg 无法解析的文件格式(wmv9, rm, rmvb 等), 可以先用别的工具(mencoder)转换为 avi 格式(ffmpeg 能解析的), 原理和上面的处理 processFLV() 方法是一样的。下面展示的是转换成 AVI 格式的 processAVI() 方法的代码片段:

```
List<String> commend = new java.util.
ArrayList<String>();
```

```

commend.add(softAVIPath);
commend.add(this.oldFilePath);
commend.add("-oac");
commend.add("-lavc");
commend.add("-lavcopts");
commend.add("acodec=mp3:abirrate=64");
commend.add("-ovc");
commend.add("xvid");
commend.add("-xvidencopts");
commend.add("bitrate=600");
commend.add("-of");
commend.add("avi");
commend.add("-o");
commend.add(this.newFilePathAVI);

```

通过上传以及解码器解码,将视频文件转换成 flv 文件,并通过内嵌于网页的 flash 播放器播放的 flv 文件的效果图。

### 2.3 提高系统性能的算法

在处理视频文件转换的时候,将会遇到很多性能问题,比如并发运行多个 ffmpeg 将会占用很大 CPU 使用,最终会导致服务器死机瘫痪。因此,Web 后台转换视频格式时采用队列的数据结构,而不是采用并发的方式。一次只转换 1 个视频文件,并且要考虑到注意线程阻塞问题。下面的伪码是解决阻塞问题的算法。

```

if (!IsConverting)
{
    Thread ConvertingThread = new
Thread(new ThreadStart(Converting));
    ConvertingThread.start();
}
// 转化视频
private void Converting()
{
    IsConverting = true;
    mutex.wait();
    while (select top 1 from XXX
where converted = 0)
    {
        // 取出一个未转化的视频
        ConvertVideo(unConver-
tedVideo);
    }
}

```

```

}
mutex.stop();
IsConverting = false;
}

```

## 3 结束语

该播客技术及视频广告技术已经成功应用于一个电子商务平台,在开发相应的模块时,相似的代码可以复用,当然视频广告模块还有更为复杂的竞价的业务逻辑。后台程序通过解码器将各种视频文件转换成 flv 格式,再调用前台网页的 flash 视频播放器,将 flv 文件嵌入 flash 视频播放器中播放,它比传统的视频播放有更好的效果及画质。如果需要进一步地开发在线视频录制,也只需在服务器端安装 Flash Media Server,客户端调用其远程服务即可。播客及视频广告系统最重要的还是性能的问题,当大量地用户上传视频文件时,一台服务器将会显得远远不够,可以考虑使用服务器群集以及采用负载均衡的策略。同时还得考虑处理上传大文件的问题。在处理视频文件转换时,还有更进一步的方法,即采用任务调度的方式,定期检测是否有待转换的文件,有的话通过队列进行转换,直到 ffmpeg 进程结束为止。然后下一个队列的转换,否则就无限等待。

### 参考文献:

- [1] Bruce Eckel. Java 编程思想[M]. (第2版) 北京: 机械工业出版社, 2002.
- [2] 唐森, 赫鲁. expert one-on-one J2EE Development without EJB[M]. 北京: 电子工业出版社, 2005.
- [3] Michael Geoghegan, Dan Klass Podcast Solutions (friends of ED, 2005), ISBN 1-5905-9554-8.
- [4] flvtool2 reference: <http://rubyforge.org/projects/flvtool2/>
- [5] 戎伟, 孟勤, 苏威. 揭秘 J2EE 项目开发的 70 个问题[M]. 北京: 人民邮电出版社, 2007.
- [6] 朱爱宇. 基于 .NET 架构的影音网站开发[D]. 山东大学硕士论文, 2006.
- [7] 王跃云. Struts 框架视频服务管理研究[D]. 太原理工大学硕士论文, 2006.
- [8] 吕辉. Flash/Flex ActionScript 3.0 交互式开发详解: 语法基础、典型应用、视频网站实战[M]. 北京: 电子工业出版社, 2008.