

# 新媒体广播服务器的设计与实现<sup>①</sup>

董子强<sup>1,2</sup>, 李鸿彬<sup>2</sup>, 杨雪华<sup>1,3</sup>

<sup>1</sup>(中国科学院大学, 北京 100049)

<sup>2</sup>(中国科学院沈阳计算技术研究所, 沈阳 110168)

<sup>3</sup>(沈阳师范大学, 沈阳 110034)

**摘要:** 随着互联网技术的发展,传统的广播电台行业受网络媒体的影响越来越大. 通过对广播行业的调查研究发现,传统广播正向新媒体广播开始转变. 针对现有的流媒体服务器不具有多种音频源自动切换,控制功能薄弱,播放延迟大等问题,设计并实现了一个符合新媒体要求的新媒体广播服务器. 介绍了流媒体音频服务器的应用及研究现状,阐述了广播行业对新媒体广播服务器的具体要求,描述了新媒体广播服务器的功能,详细分析了环形缓冲线程同步模型、音频源自动切换方法,远程控制方法. 最后通过实验测试了服务器的功能和性能,验证了新媒体服务器已满足新媒体广播需求,达到了商用目的.

**关键词:** 新媒体; 流媒体服务器; 音频广播; HLS; Jack

## Design and Implementation of a New Media Server

DONG Zi-Qiang<sup>1,2</sup>, LI Hong-Bin<sup>2</sup>, YANG Xue-Hua<sup>1,3</sup>

<sup>1</sup>(University of Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(College of Software, Shenyang Normal University, Shenyang 110034, China)

**Abstract:** With the development of internet technology, traditional radio stations are affected by the great influence from web media. After investigation and research on broadcasting industry, traditional radio begins to shift towards new media broadcasting. The current streaming media server fails to switch automatically among multiple audio sources. Furthermore, it is insufficient in control functions and broadcast delay. A new media broadcasting server, against the deficiencies of the current streaming media server mentioned above, is implemented which meets the requirements of new media. We firstly introduce the current researches and applications of streaming audio server and illustrate the specific requirements of the broadcasting industry to the new media broadcasting server. Then we describe the functions of the new media broadcast sever and analyze circular buffer method for thread synchronization model, automatically switching the audio source and remote control method in detail. Finally, the functions and performance of the server are evaluated with experiments. The experimental results show that new media sever reaches the user needs and commercial purposes.

**Key words:** new media; streaming audio server; audio broadcast; http live streaming; jack

## 1 概述

随着网络的迅猛发展,传统的文字和图像已经不能满足人们对信息获得的要求. 流媒体技术的出现恰好解决了当前的问题. 流媒体以其娱乐性强、生动、直观性好等优点,广泛用于网络电视直播、视频点播、

远程教育、新闻发布、网络广告、远程医疗、视频会议等网络信息服务的各个方面,对人们的生活产生了深远的影响<sup>[1]</sup>.

目前流媒体技术<sup>[2]</sup>主要表现在三方面: 编码技术、播放器和流服务器<sup>[3]</sup>. 一个流媒体服务器要解决从硬

① 基金项目:辽宁省教育科学“十二五”规划课题(JG14DB385)

通讯作者:李鸿彬 E-mail:59968260@qq.com

收稿时间:2014-10-17;收到修改稿时间:2014-12-10

件设备采集、编码转换形成媒体流、通过各种协议传输到网络 and 用户播放等各方面问题。传统媒体的传播技术是完全下载一个音视频节目再播放, 由于媒体文件一般体积较大, 在网络带宽有限的情况需要很长的时间等待文件传输。采用流媒体技术可以实现流式传输, 将多媒体由服务器流化成连续的、一系列、不间断传送的小数据。用户不用等待媒体文件完全下载就可以播放实现了边看边下载。

流式传输主要分为顺序流传输和实时流传输两种<sup>[4]</sup>。顺序流式传输中文件可以顺序下载, 但是用户的观看和服务器传输是异步方式, 用户需要一段时间过后才能看到服务器传输的信息, 用户只能观看已经下载的部分, 不能够快进到未下载部分播放<sup>[5]</sup>。代表协议有 HLS 协议。实时流传输协议中, 用户可以快进快退, 对网络的要求比较高, 如果网络传输状况不好, 会影响播放效果。代表协议主要有 RTP、RTSP<sup>[6-8]</sup>。

新媒体广播对流媒体服务器的要求比传统的服务器更多, 要求具有远程控制、资源自动调度、节目编排、节目录音接口、可动态扩展等功能。

## 2 技术背景

由于移动网络的发展使得客户终端多样, 网络环境复杂多变, 根据这些特点我们选择 HLS 协议实现流媒体传输<sup>[9]</sup>。

### 2.1 新媒体

新媒体是相对于传统媒体而言, 是报刊、广播、电视等传统媒体以后发展起来的新的媒体形态, 是利用数字技术、网络技术、移动技术、通过互联网、无线通信网、有线网络等渠道以及电脑、手机、数字电视机等终端, 向用户提供信息和娱乐的传播形态和媒体形态。

### 2.2 HLS 协议

HLS, Http Live Streaming 是由 Apple 公司开发的基于渐进下载的传输协议, HLS 协议是基于 Http 协议的, 因此具有防火墙穿透能力, 可为世界任意一个网络的用户提供相同的服务。可为用户从一个网络环境换到其他网络环境时可以提供不间断的服务。HLS 协议具有节能、cpu 利用率低等优点, 可以给移动互联网用户带来更好的体验。

## 3 新媒体服务器设计

### 3.1 整体架构的设计

#### 3.1.1 功能设计

新媒体广播服务器是符合现在新媒体功能要求的服务器, 要求可提供稳定可靠高品质的音频流输出, 要求具有高度的可扩展性和鲁棒性, 要求具有高度的适应性和可控性。

新媒体广播服务器的主要功能:

① 对各种声卡的音频采集工作。根据 linux 系统的稳定性和安全性, 选择在 linux 系统下实现对各种声卡进行采集工作。并且通过 jack 客户端服务器模式分离了底层和应用的耦合度。

② 编码流化。实现对采集数据的各种编码, 支持 aac, mp3, wav 等编码方式并且将所有数据按 HLS 协议的要求分成连续的 ts 段和生成 m3u8 播放文件。

③ 实现数据库节目编单和每个节目时间段内的各种声卡源和文件源动态任意切换功能。

④ 实现云平台对各个新媒体广播电台的控制功能接口, 实现频率的开播停播转播等。

⑤ 录制功能接口。实现与慢录剪辑服务器实现 tcp 通信传输数据便于录制剪辑节目。

⑥ 媒体库自动管理功能。实现音频文件媒体库的添加删除等功能, 实现节目单编排后的音频资源文件自动分配和回收删除。

#### 3.1.2 系统模块划分

新媒体广播服务器主要分为音频采集流化输出模块, 媒体库自动管理, 控制模块, 同步模块。整体结构图如图 1。

音频采集流化模块: 主要实现根据数据库节目单从音频设备和音频文件采集数据编码流化输出, 产生稳定的 ts 流。

同步模块: 此模块负责将产生的 ts 流同步到云端发布服务器。采用 TCP 协议实现与慢录剪辑服务器的交互, 将播放的内容实时传输到慢录剪辑服务器。

媒体库自动管理模块: 根据数据库中的节目单和电台频率自动同步上传的音频文件和提前分发节目单中需要的音频文件到指定的位置, 保证节目正常播放。

控制模块: 实现对音频采集流化模块的控制功能。

### 4 新媒体广播服务器实现

新媒体广播服务器是在 linux 环境下用 C++ 开发. 涉及音频文件解码位置定位算法、I/O 复用 epoll 技术、线程间的同步与通信、动态音频源的切换技术、带环形缓存区线程通信设计、控制通信命令设计, linux 文件事件监控, 文件同步算法 rsync 等.

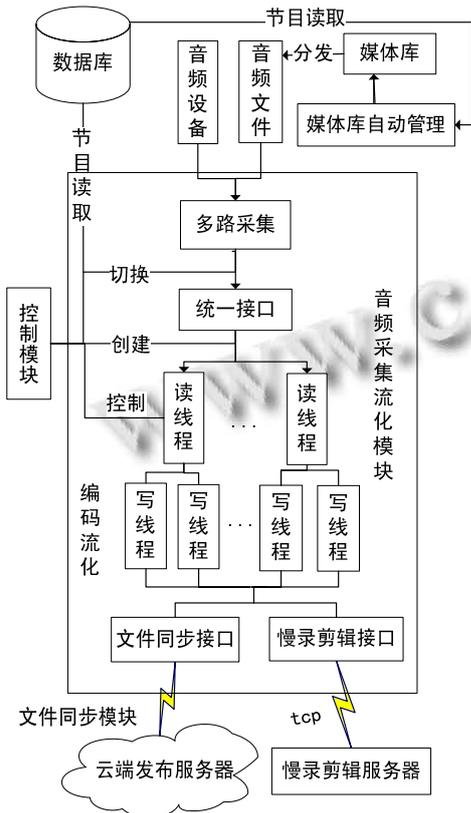


图 1 整体结构图

#### 4.1 音频采集流化模块

此模块是整个设计中最重要的重要组成部分, 在实现中采用抽象类 Source 实现了对音频源的统一接口, 采用抽象类 Sink 实现了对音频流统一的输出接口. 采用抽象类 Encoder 实现了各类编码方式的统一接口. 采用多线程, 读者写者的缓冲模型实现.

##### 4.1.1 抽象类的设计

Source 抽象类的子类包括: MediaSource 类, JackSource 类, OssSource 类等, 每个子类实现对不同的设备类型或音频文件类型进行音频数据的读取, 此设计方式用到多态的概念可以方便的设置统一的读取数据接口.

Source 抽象类部分 UML 类图如图 2.

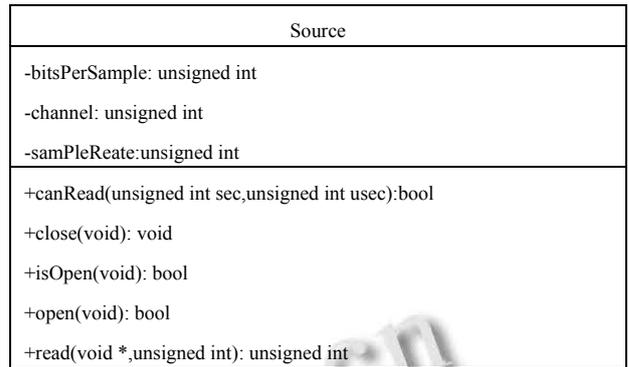


图 2 Source 抽象类 UML 图

Sink 抽象类部分 UML 类图如图 3

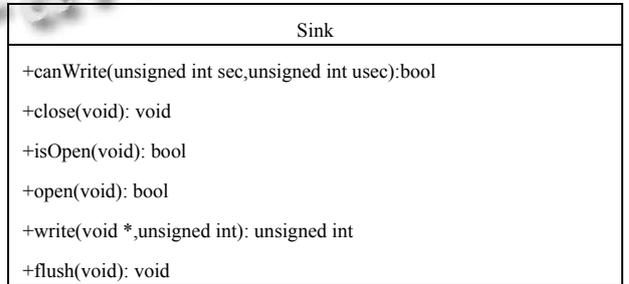


图 3 Sink 抽象类 UML 图

#### 4.1.2 环形缓冲器线程同步模型

为了解决音频文件解码线程和读线程之间的同步问题, 设计了一种带环形缓冲器的线程同步模型. 由于解码线程的解码速度远远快于解码后数据的播放速度, 当解码后的数据播放时间超过当前时间  $T$  时, 读线程主动睡眠  $T_w$  等待音频数据播放一段时间, 这样可以提高性能, 减少 cpu 和内存占用率, 保证数据解码和播放之间具有较小的延迟.

$$T = (T_p - T_f) * C / N_f \tag{1}$$

$$T_w = T - T_d \tag{2}$$

其中  $T$  为环形缓冲器满时数据播放的时间和当前系统时间的差.  $T_p$  为一帧数据的播放时间,  $T_f$  为一帧数据的解码时间,  $C$  为环形缓冲器的容量,  $N_f$  一帧数据的大小,  $T_w$  为睡眠时间,  $T_d$  为解码后数据到输出之间的延迟.  $T_d$  通过实际测试获得数据的大小.

缓冲区大小的设置根据所读文件帧的大小设定. 最小值应该大于支持文件类型的帧的最大值. 线程结束的条件为文件读完或者读取设备返回的值为 0 字节. 带环形缓冲器线程同步模型主要原理如图 4 所示.

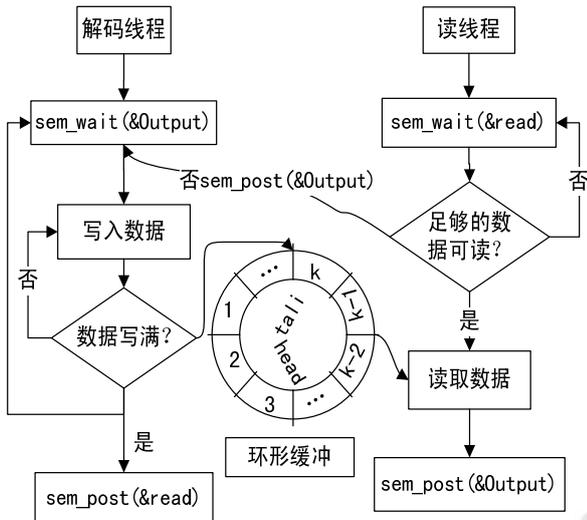


图4 环形缓冲器同步模型

4.1.3 Mp3 文件解码位置定位方法

节目中如果含有音频文件，需要根据节目的剩余时间设置音频文件解码的起始和结束位置。解码的起始位置和结束位置是指在音频文件中解码的起始帧和结束帧。对于不同参数的Mp3 音频文件具有不同的计算方法。由于计算的是帧到帧的位置，可能存在帧到帧之间的播放时间和节目剩余时间的不对应，对此采用比节目剩余时间大的帧的最小个数的方法来解决此问题。由于采样率为44100hz的Mp3 文件每一帧的大小为26ms，所以此方法误差在可接受的范围之内。

MPEG 规定每一帧的样本数为1152个样本。节目剩余时间为T，样本率为S，剩余时间内需要的帧数为N。

$$N = [T / (1152 / S)] + 1 \quad (3)$$

如果文件的总帧数小于N说明此文件不能够满足节目剩余时间的要求，需要再播放节目中下一个音频文件。

4.1.4 音频源切换方法

根据数据库节目单实现了节目时间段内部的音频源切换，可以在同一个节目中实现音频设备输入源到另一个音频输入源，设备与音频文件之间的相互切换。Mp3 音频播放时间的计算公式如下：

$$T = B / (b / 8 * c * s) \quad (4)$$

其中T为播放时间，B为转换的总字节数，b每个样本的位数，c通道数，s样本率。需要分别统计不同节目源转换的字节数然后分别根据公式(4)计算时间，当到达节

目终止时，重新读取数据库，根据数据库中的内容重新给音频源类赋值。由此可以改变音频源，实现音频源的切换。

4.2 控制模块

该模块提供一个可视化的管理接口，包括频率的添加删除开播开播，节目单的编辑，资源调度等。

其中频率的控制功能是通过Socket 线程监听控制信息然后通知相应线程处理。此函数对外接口为：

control(int ch, int type,char \* values,int length)参数详细分析如表1所示。

表1 控制参数

频率 Id	状态	启动配置	资源路径
01	1	配置文件	/Media/live/01
02	1	配置文件	/Media/live/02
03	1	配置文件	/Media/live/03

4.3 文件同步模块

该模块使用rsync 算法和linux 内核支持的inotify 特性实现。通过inotify 监控文件目录的改变，触发rsync 算法进行文件同步。此过程是内核主动触发，效率高。Inotify 监控的事件主要包括delete, create, attrib, write 等操作，需要注意的是如果要写入一个大的文件inotify 会持续的发送更新事件，并且调用同步过程进行同步。对此可以等待写结束文件之后再更新。具体设置为使用参数-e close\_write。

4.4 媒体库自动管理

该模块根据数据库节目单中的节目自动管理需要用到的音频文件资源。并且根据节目单中频率的资源路径，将用到的资源文件自动备份到路径之下。并且当节目结束时将此路径下的资源删除。

5 实验

实验在Ubuntu 12.04 环境下测试，硬件配置为Dual-Core E6600 3.06GHz，内存2G，硬盘500G。

实验对三个不同的频率进行功能测试和性能测试。表3中设置频率内节目的时间间隔分别为1小时，1分钟和6小时。同时频率间节目资源类型交叉设置用来测试各种切换。表4中播放周期中的1234567指一周中的7天。

在FreinfoList 表中添加数据如表2所示。

表 2 频率信息表

频率 Id	状态	启动配置	资源路径
01	1	配置文件	/Media/live/01
02	1	配置文件	/Media/live/02
03	1	配置文件	/Media/live/03

在表 EpginfoList 中添加数据如表 3

表 3 功能测试节目单

频率 Id	名称	开始时间	结束时间	资源类型	设备类型
01	音乐 1	08:00	09:00	设备	Rme
01	交通 1	09:00	10:00	文件	Mp3
01	经济 1	10:00	11:00	设备	Rme
02	音乐 2	08:59	09:00	设备	Rme
02	交通 2	09:00	09:01	设备	Rme
02	经济 2	09:01	09:02	文件	Mp3
03	音乐 3	08:00	12:00	文件	Mp3
03	交通 3	12:00	16:00	文件	wav
03	经济 3	16:00	20:00	设备	Rme

表 4 性能测试节目单

频率 Id	开始时间	结束时间	播放周期	资源类型	设备类型
01	00:00	08:00	1234567	设备	Rme
01	08:00	16:00	1234567	文件	Mp3
01	16:00	24:00	1234567	设备	Rme
02	00:00	09:00	1234567	设备	Rme
02	09:00	10:00	1234567	设备	Rme
02	10:00	24:00	1234567	文件	Mp3
03	00:00	12:00	1234567	文件	Mp3
03	12:00	15:00	1234567	文件	wav
03	15:00	20:00	1234567	设备	Rme

实现结果主要考虑以下几方面，模块对控制命令的响应时间，包括频率启动、停播、转播、节目切换的时间延迟，实验结果如图 5 所示。性能测试主要考虑 cpu、内存占用率和稳定性<sup>[10]</sup>。通过连续 15 天测试，每天取三个样本点求均值表示一天内数据。实验结果如图 5 所示。

实验结果显示转播命令的延迟最大，但延迟在 20ms 以下。由于人耳分辨两个声音的间隔为 0.1 秒，所以 20ms 延迟完全可以满足广播需求，通过 15 天的测试稳定性良好，开启三个频率时 cpu 占用率在 29% 左右，内存 22% 左右。

实验表明新媒体服务器功能和性能可以满足新媒体广播的要求。

## 6 结语

本文通过对新媒体广播现状的分析研究，设计并实现了一个符合要求的新媒体广播服务器。首先分析

了新媒体广播服务器的作用，接着详细描述了新媒体服务器的设计和实现过程，对部分抽象类和过程做了详尽分析，实现了新媒体服务器的预期功能，后续仍需要进一步完善各个模块，提高性能，使新媒体广播服务器在新媒体广播中有更好的应用。

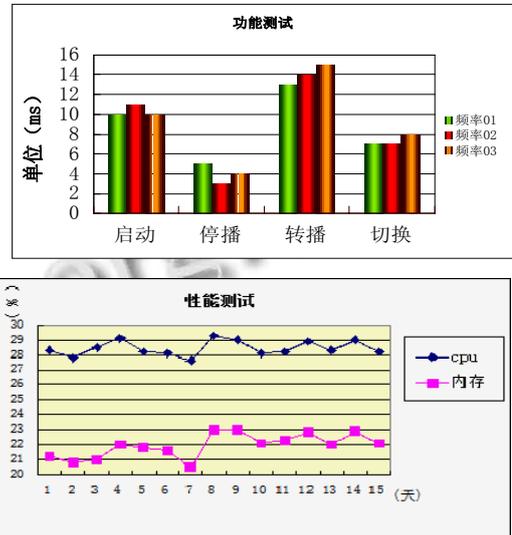


图 5 实验结果

## 参考文献

- 1 贾代平,范辉.流媒体技术规范及其应用技术研究综.计算机工程,2005,31(2):5-6.
- 2 陈爽文.流媒体技术综述.北京广播学院报,2003,10(1):58-64.
- 3 Nakanishi H, Mamoru K. Design for real-time data acquisition based on streaming technology. Fusion Engineering and Design, 2001,56:1011-1016.
- 4 李建廷.流媒体技术综述.现代电子技术,2005,28(3):91-93.
- 5 Massoulié L, Twigg A. Rate-optimal schemes for Peer-to-Peer live streaming. Performance Evaluation, 2008,65(11):804-822.
- 6 范俊辉,田斌,陈强.流媒体实时传输技术的研究.电子科技, 2005,(12):42-45.
- 7 Gruber S, Rexford J, Basso A. Protocol considerations for a prefix-caching proxy for multimedia streams. Computer Networks, 2000,33(1):657-668.
- 8 Kuschnig R, Kofler I, Ransburg M. Hermann Hellwagner design options and comparison of in-network H.2-64/SVC adaptation. Journal of Visual Communication and Image Representation, 2008,19(8):529-542.
- 9 李晓城,钱松荣.一种自适应的 3G 网络流媒体速率控制算法.小型微型计算机系统,2012,33(7):1429-1432.
- 10 李学森,李俊奚,宏生.流媒体服务器性能评价模型.计算机工程,2009,35(10):270-272.