

面向新媒体广播的慢录剪辑服务器^①

刘 利^{1,2}, 李鸿彬², 杨雪华^{1,3}

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

³(沈阳师范大学, 沈阳 110034)

摘 要: 随着流媒体技术的飞速发展和完善, 传统流媒体广播服务器已经不能满足大众们的收听需求. 为提高传统流媒体服务器的整体性能, 通过对传统流媒体服务器功能和结构的分析研究, 设计和实现了慢录剪辑服务器系统, 将慢录剪辑功能从传统流媒体服务器中分离出来. 通过采用模块化设计方法, 设计并实现各相应功能模块, 确保了对媒体流的实时接收, 完成了对接收的媒体流进行正确的剪辑、同步备份, 实现了慢录剪辑服务器工作的高稳定性和同步的及时性. 采用以理论分析和实验论证相结合的方法, 表明所设计慢录剪辑服务器的工作稳定性和有效性均满足要求.

关键词: 流媒体技术; 实时; 慢录剪辑; 同步; 备份; 模块化

Recording Clip Server for New Media Broadcasting

LIU Li^{1,2}, LI Hong-Bin², YANG Xue-Hua^{1,3}

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(College of Software, Shenyang Normal University, Shenyang 110034, China)

Abstract: With the development and improvement of the streaming media technology, the traditional streaming media server has been unable to meet the needs of public. In order to improve the capability of the streaming media server, we designed and implemented a recording clip server, which is separated from the traditional streaming media server with analysis the function and structure of traditional streaming media server. We use the modular design approach to implement the corresponding function modules. It ensured that the real-time received of media stream, the right clips and backups of media stream and worked high stable and synchronization. This paper intends to design a recording clip server which could work stable and have good performance by the combination of theory and experiment.

Key words: streaming media technology; real-time; recording clip server; synchronization; backups; modular design approach

近年来随着互联网的快速发展, 流媒体应运而生并得到了迅猛的发展. 尤其是在移动互联网飞速发展的今天, 新媒体正在悄然的改变用户的收听习惯^[1]. 传统广播以其老旧的体系和功能结构已然成为了“落后媒体”“老旧媒体”, 其在本世纪的持续发展正面临着巨大的挑战^[2]. 探索出一条既能保持自己的优势, 又能够通过采用对现有的节目内容、传播媒介、商业模式进行改造, 实现构建快速度、开放式、个人化电台

成为了摆在人们面前的重要课题.

随着流媒体技术的飞速发展和完善, 尤其是移动智能设备软硬件不断推陈出新, 在继承和发扬广播媒体的传统优势的前提下, 传统广播开始向流媒体平台拓展业务. 流媒体广播除提供广播节目的实时收听外, 还提供伴随广播节目的新闻娱乐咨询、互动游戏、社区交友等服务^[3]. 由于广播节目和互动服务内容的实时性、快捷性、多变性以及再需求性^[4], 如何在提升直播流媒体服务器的性能的前提下, 保证对广播内容的按时剪切、录音以及及时备份再次播放的完整性有了

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: 2014-10-15; 收到修改稿时间: 2014-12-01

更高的要求. 实现对流媒体广播服务器的流内容准确、实时的备份, 具有重要的理论研究意义和实际应用价值^[5].

目前, 市场上主流的流媒体服务器平台有三种, 分别是: RealNetwork 公司的 Real System Server、Microsoft 的 Windows Media Server 和 Apple 公司的 QuickTime Streaming Server. 其慢录剪辑功能的实现依赖于第三方软件或内嵌入整体系统中, 绝大多数均将其作为流媒体服务器的补充功能, 并且不能做到自动化剪辑录音. 本文针对大众对流媒体服务器的新需求、新标准: 设计并实现了慢录剪辑服务器, 对比现有的流媒体平台架构^[6], 特别将录制、剪辑功能从流媒体服务器分离出来, 减少了直播服务器运行录制、剪辑功能时对于系统资源的消耗. 除了保证在直播时按要求自动无损、准确的备份播出内容, 实现了直播音频源与用户的互动直达, 而且更好的保证了多播条件下流媒体服务器的准确性、稳定性.

1 系统概述

1.1 整体结构

慢录剪辑服务器模块分为: 参数接口(Exported Parameters)和控制中心, 主要用于在配置文件里设置参数; 数据库通信接口, 主要用来获得慢录剪辑信息; 与音频矩阵服务器通信接口, 用于建立通信模式, 并实时获得音频数据流; 与媒体库服务器通信接口, 主要实现对录音文件的实时同步; 慢录剪辑模块, 根据已经获得的音频流和慢录剪辑信息完成录音; 监控重新启动服务模块, 实现了慢录剪辑服务器工作的稳定性和高效性. 慢录剪辑服务器的整体结构如图 1 所示.

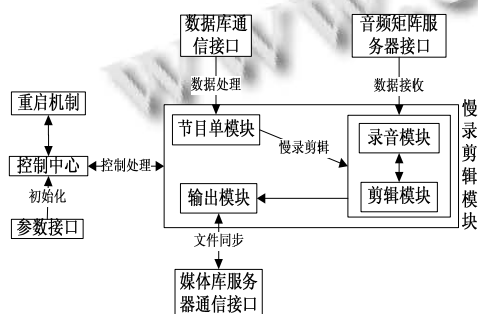


图 1 慢录剪辑服务器整体结构图

2 慢录剪辑服务器的设计

2.1 功能设计

慢录剪辑服务器在广播电台内系统中主要完成对需要录音或备份的音频资源进行准确的备份, 确保在任何时候都可以进行录音和剪辑, 同时完成对已录文件进行查看和再次播放, 慢录剪辑服务器的主要功能如下:

1) 通过与数据库服务器进行通信, 获得所需要的慢录剪辑信息, 包括慢录文件存储目录、剪辑信息和时间要求等.

2) 通过与音频矩阵服务器通信, 根据已经获得的慢录剪辑信息, 通过实时监控通信端口, 从而完成录音和按要求剪辑.

3) 通过同步文件系统, 将录制好的音频文件同步到媒体库服务器, 用于作为将来的播出资源.

慢录剪辑服务器的主要编程语言是 C 语言, 操作系统选择了工作稳定性和可靠性均很高的 Linux 操作系统. 根据相应的功能需求设计并实现各个子模块.

2.2 慢录剪辑系统实现机制

慢录剪辑服务系统主要工作对象是来自音频矩阵服务器的不同频率的音频流, 因此存在多频率音频流同时录音、剪辑的时间, 同时对各线程的控制依赖于音频流信息, 其中音频流信息存储在内存中, 所以慢录剪辑服务器的实现依靠的是多线程和内存共享, 其实现模型如图 2 所示.

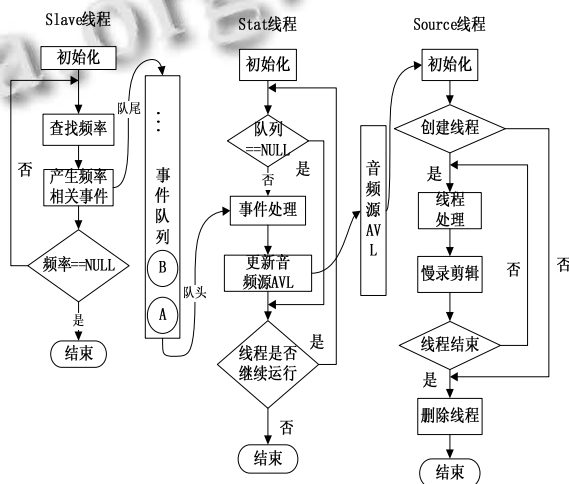


图 2 慢录剪辑服务器设计模型图

Slave 线程主要完成对网络通信端口的监听, 分析所获得的数据, 根据频率的信息生成事件并加入事件队列. Stat 线程则主要处理事件队列并更新音频源自平衡二叉树. Source 线程则根据该二叉树产生、删除相应的音频源线程以方便慢录剪辑的开始和停止. 同时为了保证线程通信的安全性, 本文采用互斥锁方法.

2.2 参数接口的实现

在慢录剪辑服务器的参数接口模块里需要设置如下几类参数, 其中参数类型如图 3.

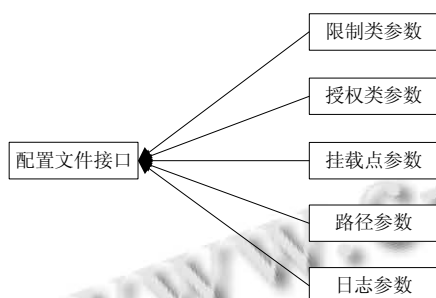


图 3 慢录剪辑服务器参数设计图

通过参数接口, 获得慢录剪辑服务器的各项基本参数; 包括慢录剪辑服务器对外的限制和授权, 挂载中音频的分布, 文件的存储路径, 日志文件的配置, 保证了整个慢录剪辑服务器的正确初始化, 极大提高了程序的可扩展性.

2.3 通信接口的设计实现

2.3.1 与数据库服务器通信接口的实现:

该接口主要实现与 MySQL 数据库服务器通信, 并从数据库中获得慢录剪辑信息. 慢录剪辑服务器运行期间, 需要定时向数据库获得相关信息, 也就是需要多次与数据库进行连接的建立与释放, 为了提高用户数的同时访问数量, 降低数据库服务器压力, 提高数据库性能和减少整个系统的各项资源开销, 我们设计了数据库连接池作为慢录剪辑服务器与数据库服务器进行通信的接口^[7], 其基本设计原理如图 4 所示.

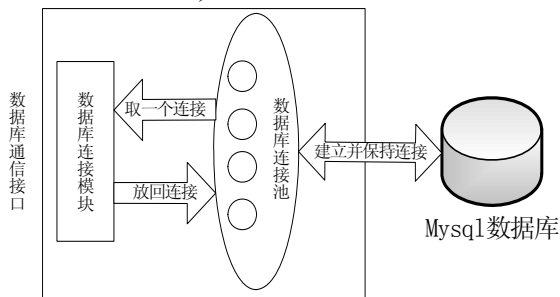


图 4 数据库通信接口设计图

采用如图 4 数据库连接方法时, 数据库连接模块向数据库连接池请求获取连接, 当数据库连接池有该数据库的链接时则让其获取连接, 否则数据库连接池会与数据库建立连接并将连接返回给数据库连接模块, 在获取到慢录剪辑信息后, 并不是释放数据库连接, 而是把它交给数据库连接池保管, 以便下次调用.

同时通过节目单模块将从数据库服务器提取出来的慢录剪辑信息根据各自不同的节目单生成相应的慢录剪辑链表, 完成对当天的节目进行录音、剪辑.

2.3.2 与媒体库服务器通信接口的实现:

慢录剪辑服务器与媒体库服务器通信的主要功能是实现将慢录剪辑服务器成功剪辑并录音的文件同步到媒体库服务器备份起来, 考虑到要保证备份的准确性和有效性, 采用已有的成熟项目文件同步系统(NFS)来设计与媒体库服务器的通信接口, 慢录剪辑服务器作为 NFS 客户端, 媒体库服务器作为 NFS 服务器, 通过调用远程调用协议(RPC), 实现将 NFS 服务器端设置好共享权限的目录挂载到 NFS 客户端, 实现挂载目录下的文件同步^[8].

2.3.3 与流媒体音频矩阵服务器通信接口的实现:

流媒体音频矩阵服务器是慢录剪辑服务器的音频源, 为了保证对音频流的实时监控, 以 Sockets 通信方案为基础, 实现了与流媒体音频矩阵服务器的通信接口^[9]. 同时, 为了保证音频流的准确性和正确性, 通信协议采用 HTTP 协议. HTTP 协议包括请求报文、响应报文、普通报文^[10]. 通过该三种报文机制, 使得慢录剪辑服务器与流媒体音频矩阵服务器之间建立了完整的请求、应答和数据传输机制. 通过该方法, 可以实时获取该端口的数据, 并进行分析和处理, 从而将音频数据进行准确、准时的备份.

为了保证能够同时对多个不同频率节目进行录音、剪辑, 需要对通过端口侦听到的数据进行分析, 查找出相应的频率信息以完成对频率在系统中的挂载, 其中对频率及其相关信息的存储, 采用自平衡二叉树的方法, 其设计模型如图 5 所示.

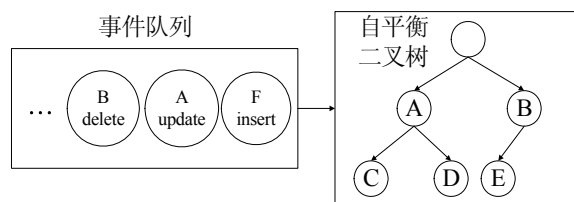


图 5 音频源信息存储模型图

通过该模型,可以得到一个存储音频源信息的自平衡二叉树,其中每个节点存储音频源的各个频率信息、相关配置信息和运行信息.慢录剪辑系统则根据该自平衡二叉树除头节点以外的节点生成相对应的线程,通过该线程完成对节目的慢录剪辑.

2.4 慢录剪辑模块的设计实现

慢录剪辑模块是慢录剪辑服务器的最核心部分,它的主要功能是根据我们已经获得的慢录剪辑信息将音频数据进行剪辑和备份.主要设计模型如图6所示.

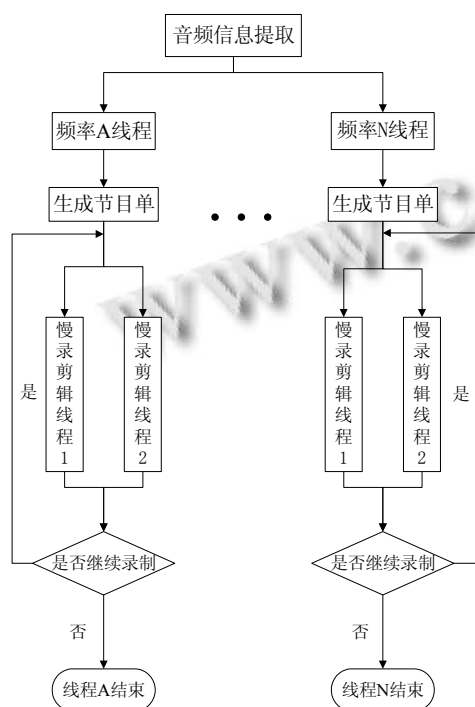


图6 慢录剪辑模块设计模型图

如图6所示,对音频信息的提取通过存储音频源信息的自平衡二叉树完成.根据每一个节点的频率和其运行状态等相关信息,创建或结束该频率的线程.与此同时,根据节目单模块生成每一个频率自己的慢录剪辑节目单.同时,本文采用双线程慢录剪辑方法,有效的避免了当同一节目单的两个相邻节目出现的如下问题:

First_endtime:第一个节目结束时间;

Second_starttime:第二个节目开始时间;

Time: 保证录音完整性,向前提数秒或向后延伸数秒;

$$Second_starttime - First_endtime < Time \quad (1)$$

同时,我们将根据满足慢录剪辑信息节目单,另两个子线程分别读取节目单的前一个节目和后一个节

目,有效的保证了数据慢录剪辑的完整性、准确性.

2.5 监控启动服务模块

监控启动服务模块,独立于慢录剪辑服务系统之外,其主要作用是监视慢录剪辑服务系统的运行状态,确保了慢录剪辑服务器的工作的高稳定性和有效性.

3 实验测试

在本文的测试阶段,实验环境为一台慢录剪辑服务器、一台媒体库服务器和一台音频矩阵服务器,三台服务器均安装Linux操作系统,分别安装相应的服务系统并进行通信测试.硬件环境均为: Dual-Core E5300 2.60GHz,内存 2G,硬盘 500G.需要添加节目资源信息表,其中表中主要信息如表1所示.

表1 节目资源信息表

节目名	开始时间	结束时间	开播	录音	频率
笑话集	00:00:00	18:40:00	1	1	101
音乐会	18:40:00	24:00:00	1	1	101
新闻波	00:00:00	07:30:00	1	1	102
早餐报	07:30:00	24:00:00	0	1	102

开始时间和结束时间均是节目的播放和录制、剪辑时间,开播字段代表着该节目是否属于播放节目,录音字段代表该节目是否按照开始时间和结束时间进行剪辑、录音,而频率字段表明该节目属于电台的哪一个频率.通过测试,我们得出实验结果,即慢录录音文件表 *Y_SRResList*,表2所示为部分实验测试结果.

表2 慢录录音文件表 2014_SRRexList

存储路径	文件名	大小	创建时间
/media/records/ 102/2014	2014051800:00:00 _新闻波.mp3	60.4GB	2014-05-18 07:30:16
/media/records/ 101/2014	2014051800:00:00 _笑话集.mp3	80.5GB	2014-05-18 18:40:16
/media/records/ 101/2014	2014051818:40:00 _音乐会.mp3	50.0GB	2014-05-18 00:00:16

实验结果表明,经过长时间持续运行,慢录剪辑服务器存储的音频文件被同步到媒体库服务器指定目录下,并发现实验结果均满足测试要求,表明该慢录剪辑服务器在持续运行性能上已经满足要求.同时测试已录录音文件质量,测试结果如图7所示.

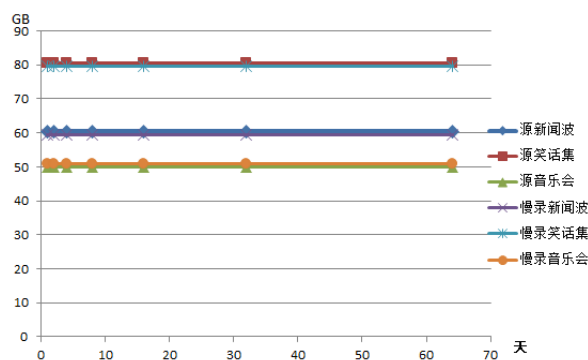


图 7 慢录文件质量测试图

源音频文件在经过长时间按的录制之后,大小并没有任何减少,并通过反复收听,音频文件的开始内容和结束内容相同,表明慢录剪辑服务器录音文件的准确性和质量均满足要求。下图 8 所示为慢录剪辑服务器在持续运行时的 CPU 在单线程、四线程和十六线程的每天的平均占用率及内存(Mem)占用率。

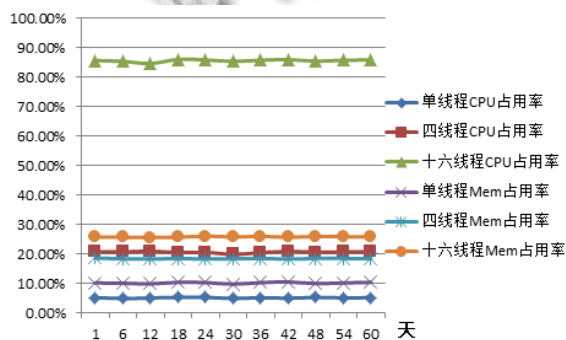


图 8 CPU 和内存平均占用率图

实验结果表明 CPU 在十六线程下处于繁忙状态,这在实际应用中是不利的,因此必须要减少对于 CPU 的平均占用率。通过对慢录剪辑服务器的整体分析,我们发现在对图 6 中音频线程和其中慢录剪辑线程的处理不够谨慎。首先是对于音频线程而言,应在不接受音频资源时尽早准确的关闭该线程有助于减少资源的消耗,而不是仅仅等待连接断开。我们重新设计音频线程的运行条件,包括增加线程关闭条件,包括是否在一定时间内接受到数据、是否出现连续无用数据等,争取尽早关闭无用的音频线程。其次对于慢录剪辑线程而言,不需要时时刻刻保证双线程的慢录剪辑,如果没有出现连续节目需要录音的情况,我们只需要单线程录音,并且可以采用轮转的方法依次创建和结束线程,进一步减少资源的消耗。

下图 9 所示为慢录剪辑服务器调整后在持续运行时的 CPU 在单线程、四线程和十六线程的十一个时间段内的平均占用率及内存(Mem)占用率。

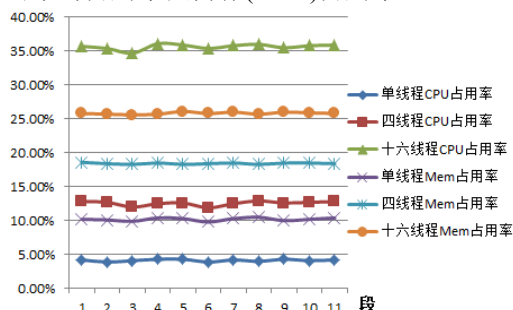


图 9 CPU 和内存平均占用率图

检测表明慢录剪辑服务器运行状态良好, CPU 和 Mem 在十六线程下处于清闲状态,相比于之前 CPU 利用率减少了一半以上,主要是由于及时关闭了长时等待音频线程,并关闭了多余慢录剪辑线程。服务器运行状态良好,运行结果准确,证明该慢录剪辑服务器拥有良好的性能和价值。

4 结语

本文通过对 Icecast、Ices 等开源代码的分析研究,设计并实现了一个慢录剪辑服务器。首先分析了慢录剪辑服务器在广播内部系统中的作用,接着详细描述了慢录剪辑服务器的各个模块和接口的设计和实现过程,对各个模块进行了详尽的分析,逐步完成慢录剪辑服务器的各个功能。后续仍需不断的优化慢录剪辑服务器系统,减少对 CPU 的占用率,不断提升慢录剪辑服务器的工作高效性。

参考文献

- 1 余芳庭.VCR 流媒体服务器的研究与设计[硕士学位论文].武汉:武汉理工大学,2008.
- 2 张宗国.基于多数据子流的 P2P 流媒体直播客户端的设计与实现[硕士学位论文].成都:电子科技大学,2013.
- 3 张彩,曹璐.新媒体技术背景下美英日“重塑广播”的新态势.河南社会科学,2011,19(5):122-124.
- 4 黄国伟,陈志.P2P 流媒体数据块的优化调度策略.计算机工程,2013,39(2):90-93.
- 5 Liu Y, Du B, Wang SL, Yang HB. Design and implementation of performance testing utility for RTSP streaming media server. IEEE 2010 First International Conf. on Pervasive

- Computing, Signal Processing and Applications. Harbin. IEEE. 2010. 193–196.
- 6 Rangaswami R, Dimitrijevic Z, Chang E. MEMS-based disk buffer for streaming media servers. IEEE Proc. of the 19th International Conf. on Data Engineering. University of California, Santa Barbara. IEEE. 2003. 619–630.
- 7 Feng GL, Yang LH. A new method in improving database connection pool model. World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering, 2007, 1(5): 1547–1550.
- 8 Bhat WA, Quadri SMK. Understanding and Mitigating Security Issues in Sun NFS[Thesis]. Srinagar, Indian: University of Kashmir, 2013.
- 9 Mohamed N, Al-Jaroodi J. Self-configured multiple-network-interface socket. Journal of Network and Computer Applications, 2009, 33(2010): 35–42.
- 10 Callegari C, Giordano S, Pagano M. Behavior analysis of TCP Linux variants. Computer Networks, 2011, 56(2012): 462–476.