

# 网络广播流媒体发布系统<sup>①</sup>

陈平<sup>1,2</sup>, 林 浒<sup>2</sup>, 杨海波<sup>2</sup>, 贾正锋<sup>2</sup>, 于 飞<sup>2</sup>, 卜立平<sup>2</sup>

<sup>1</sup>(中国科学院 研究生院, 北京 100049)

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

**摘 要:** 为应对大量用户同时访问网络广播流媒体服务的需求, 流媒体发布系统至关重要. 结合网络广播系统的特点, 设计并实现了一套用于网络广播的流媒体发布系统, 该系统分为媒体流的 Relay 和流媒体的发布, 媒体流的 Relay 通过在 Darwin Streaming Server 添加中继模块, 实现媒体流的中继. 流媒体的发布通过搭建后台管理系统、前端发布平台、终端三部分来实现网络广播业务数据的发布展示. 最后, 通过多终端同时访问发布系统对发布系统进行功能测试, 可实时流畅的的浏览收听网络广播节目.

**关键词:** 网络广播; Darwin Streaming Server; 流媒体; 发布

## Webcast Streaming Media Publishing System

CHEN Ping<sup>1,2</sup>, LIN Hu<sup>2</sup>, YANG Hai-Bo<sup>2</sup>, JIA Zheng-Feng<sup>2</sup>, YU Fei<sup>2</sup>, BO Li-Ping<sup>2</sup>

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

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China)

**Abstract:** In response to the large number of users simultaneously access the webcast streaming service, streaming media publishing system is essential. In this paper, according to the characteristics of the network broadcasting system, we design and set for network broadcast streaming media publishing system, the system is divided into the Relay of media streaming and the publishing of streaming media. Through adding relay module to Darwin Streaming Server to achieve media streaming relay. Through structuring back-platform management system, front-end platform and terminal platform three parts to achieve streaming media's publishing. Finally, multi-terminal access publishing system to verify the function of the system, achieving the network broadcast media streaming and streaming media's publishing, allowing users to listen to the program more convenient.

**Key words:** Webcast; Darwin Streaming Server; streaming media; publish

## 1 引言

随着 3G 网络的快速部署、WIFI 热点城市的增多, 人们慢慢习惯使用移动设备如 3G 手机、平板电脑等获取信息. 在强大的移动互联网设备面前, 传统的借助 AM/FM 频段的收音机提供的信息量和娱乐性是无法与之匹敌的, 所以广播电台根据自身特点, 结合当前互联网、移动互联网技术和业务发展趋势推出了一项新业务: 网络广播<sup>[1]</sup>.

网络广播将广播电台的传播途径从传统的 AM/FM 方式, 拓展到互联网、移动互联网中, 它将传

统的单向广播变成双向互动交流收听. 在收听的同时, 听众可以通过移动设备将声音、视频、图片、文字和主持人实时交流, 主持人也可以将听众交流的声音、视频、图片等信息推送给其他听众. 系统最大限度的整合内容资源, 发展增值业务, 通过多种渠道, 将丰富多彩的数字内容和应用传送给千家万户, 以获取更大的社会效益和经济效益.

网络广播收听用户的激增以及网络软硬件的快速发展, 用户对流媒体的需求量越来越大. 移动流媒体的介入, 也对支持流媒体服务的软件提出大量新的

<sup>①</sup> 收稿时间:2012-11-06;收到修改稿时间:2012-12-26

需求。但某一流媒体服务器的处理能力有限,为了更好的应对如此巨大的需求量,构建一个具有媒体流 Relay 功能的发布服务器,进行媒体流中继<sup>[2,3]</sup>,扩大流媒体服务器的处理能力是最有效快捷的方法。同时针对网络广播业务的特点,让用户更好的浏览节目,搭建良好的流媒体发布系统来友好的为用户导航已是十分必要的。

本文结合网络广播业务的需求,搭建了一个媒体流 Relay 和流媒体发布的系统,并通过了多终端同时访问发布系统验证了流媒体发布系统的功能。

## 2 发布系统研究现状

### 2.1 媒体流发布

目前流媒体领域,主要有三种平台: RealNetworks 公司的 RealSystem, 微软公司的 Windows Media, 和苹果公司的 Quiktime. 三大平台都有相应的流媒体服务器作为其应用的核心。其中苹果公司的流媒体服务器 QuikTime Streaming Server(QTSS)<sup>[4]</sup>支持多个操作系统,是当前同类产品中支持平台最多的一个。且苹果公司作为 Internet 流媒体联盟的核心成员不但其流媒体的技术与解决方案符合标准,且 QuikTime Streaming Server(QTSS, Darwin Streaming Server, DSS)是完全开源的,开发人员可以对其功能进行扩充甚至不用修改服务器的核心功能<sup>[5]</sup>。因此我们选择 DSS 流媒体服务器来实现媒体流的发布。

为了高效地发布媒体内容,人们提出了内容分发网络(CDN)。CDN 技术采用分布式缓存/复制、负载均衡、流量工程和客户端重定向等技术基于 Internet 构筑一个地理上分布的内容分发网络将信息资源想网络边缘推近,用户可以在“最近”的位置快速访问到所需的内容。但是流媒体与传统静态媒体相比容量更大、时间敏感度更高、交互性要求更高。为此我们提出架设媒体流发布服务器,本系统通过在 DSS 中添加媒体流 Relay 模块来更好的实现流媒体的传输与发布。

### 2.2 流媒体发布

目前 iOS 和 Android 系统的浏览器都是 webkit 核心的,需要开发移动 Web 应用来满足这块需求。目前主要的移动 Web 应用开发框架有: Sencha Touch、JQuery Mobile 以及 PhoneGap。

其中 Sencha Touch<sup>[6]</sup>框架是基于 HTML 5 的 Mobile App 框架,它可以让 Web App 看起来像 Native App。美

丽的用户界面组件和丰富的数据管理,全部基于最新的 HTML 5 和 CSS3 的 WEB 标准,全面兼容 Android 和 iOS 设备。而且 Sencha Touch 相对来说是一个重量级的移动 Web 应用框架,适合用于开发业务逻辑比较复杂的移动 Web 应用。相比 JQuery Mobile 和 PhoneGap, Sencha Touch 针对触摸屏有更丰富的 UI 设计且支持复杂交互、纯 JavaScript 搞定布局、版本较稳定。因此,我们选择 Sencha Touch 搭建流媒体发布系统。

## 3 系统整体设计

本系统由后台管理平台、前端发布平台,基于 Sencha Touch 框架搭建出的终端,媒体流发布服务器构成。系统整体设计框架如图 1:

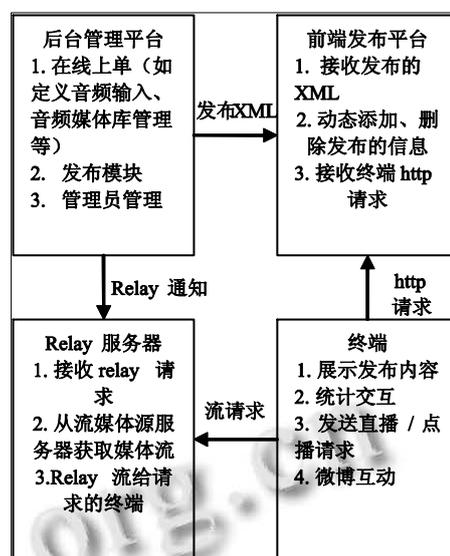


图 1 系统整体设计框架图

## 4 媒体流的Relay

### 4.1 发布服务器 Relay 功能模块设计

为了实现大量用户高质量的并发访问,我们将媒体流发送到媒体流 Relay 服务器,用户访问媒体流 Relay 服务器。这样不仅减少源流媒体服务器的负载,同时流媒体发布服务器处在高速网上,可实现最大限度的高速流媒体转发。

我们在 DSS 流媒体服务器<sup>[7]</sup>上添加了 QTSSRelay Module 模块<sup>[8,9]</sup>来实现 Relay 服务器的中继功能。DSS 媒体流 Relay 服务器在接收源服务器流媒体数据过程中充当了“客户端”角色,向流媒体源服务器请求流媒体数据,并将流媒体数据传送给缓存管理存储。同时在向用户终

端转发流媒体数据过程中充当了“服务器”角色,将缓存管理模块中相应的流媒体数据转发给用户终端,从而完成了媒体流的中继功能。

数据接收功能从流媒体源接收流媒体,并放入缓存管理,当有用户请求时,数据发送功能给用户发送数据.流媒体发布服务器 Relay 功能图 2:

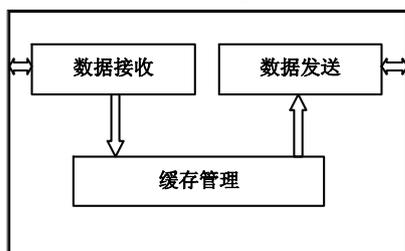


图 2 流媒体发布服务器 Relay 功能图

## 4.2 发布服务器 Relay 模块实现

Darwin 流媒体服务器架构大体上由客户端、核心服务器、服务器模块组成.核心服务器充当网络客户端和服务器模块之间的接口.当 Darwin 流媒体服务器收到一个 RTSP 请求<sup>[10]</sup>时,它会根据预先定义好的顺序,调用各个模块注册的相关角色来处理请求. Relay 服务器在不改变 Darwin 流媒体服务器核心功能的情况下,利用 Darwin 流媒体服务器提供的开发接口,将自定义模块(QTSSRelayModule)添加进去来实现流媒体中继功能。

### 4.2.1 模块的要求

有两种方法将 QTSS 模块添加到 Darwin 流媒体服务器中,一种是静态模块,直接将代码编译为服务器内核的一部分;另一种为动态模块,首先将开发的模块编译为一个动态库,在 Darwin 流媒体服务器启动时加载<sup>[11]</sup>.在保证尽可能不修改 Darwin 流媒体服务器核心代码的基础上,本文对 QTSSRelayModule 采用了动态编译的方法.每个 QTSS 模块必须实现两个例程<sup>[12]</sup>:一个是 Main 函数,服务器启动时调用这个函数对该模块进行初始化.一个是 Dispatch(分发)函数,服务器为了某种特定的目的调用模块时使用这个函数,并在调用时将任务的名称及相应的参数块传递给该函数。

QTSSRelayModule 模块的主函数体书写如下:

```
QTSS_Error QTSSRelayModule_Main(void*
inPrivateArgs)
{
```

```
Return _stublibrary_main(inPrivateArgs,
QTSSRelayModuleDispatch);
}
```

QTSSRelayModule 的分发函数原型如下:

```
void QTSSRelayModuleDispatch(QTSS_Role
inRole,QTSS_RoleParamPtr inParams);
```

### 4.2.2 模块的实现

QTSSRelayModule 主要负责 RTSP 会话的建立、RTP 数据收发.包括以下主要类:

#### (1) RTSPSession 类

抽象 RTSP 会话任务的共有数据与方法,包括该会话所控制的媒体流列表,以及发送请求,接收回复的方法等,每个 RTSP 会话都对应一个 RTSPSession 对象。

#### 重要方法

```
Public void doDescribe(String request);//负责处理
RTSP 会话中的 Describe 消息
```

```
Public void doAnnounce(String request);
```

```
Public void doSetup(String request);
```

```
Public void doTeardown(String request);
```

```
Public void doPlay(String request);
```

```
Public synchronized void findRTPPorts();
```

#### (2) StreamInfo 类

封装了每一个媒体流的信息,包括该媒体流的远端端口,本地端口,名称等。

#### (3) BufferManagement 类

负责 Buffer 对象的创建和分配,同时将中继需要的数据,根据媒体信息定位到相应的 Buffer 对象。

#### ① 重要数据结构

```
HashMap<String,Buffer> streamBuffer//媒体所对
应的缓存
```

#### ② 重要函数

```
Public Boolean Write(String meidiaName,char[]
data);//将源服务器的流媒体数据写入到缓存中
```

```
Public char[] Read(String mediaName,String
clientSession);//将缓存中读取的流媒体数据传送给相
应的客户端
```

#### (4) Buffer 类

Buffer 类是对缓存的共有数据和方法的抽象。

#### 重要函数

```
Public Boolean Write(char[] data);
```

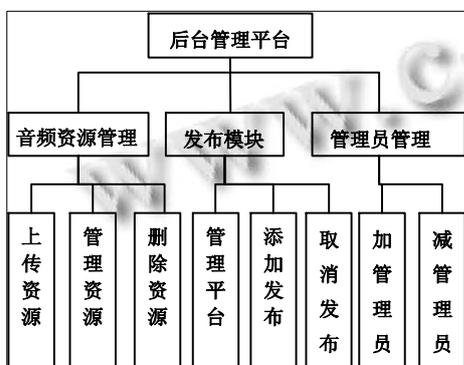
```
Public char[] Read(String clientSession);
```

## 5 流媒体的发布

流媒体的发布主要是后台将发布的媒体描述信息发布到前台并最终在 Sencha Touch 终端上展示出来的过程。其中涉及到后台媒体信息的上传、删除、存储到发布。前台对发布数据的处理和 Sencha Touch 对发布数据的展示。

### 5.1 后台管理平台

后台管理平台主要功能音频媒体库管理模块，负责音频资源的上传、删除、存储等，发布模块，负责管理前端发布平台信息，并实现将音频资源专辑以 XML 形式发布或取消发布到多个前端发布平台。后台管理平台功能如图 3:



后台管理平台系统设计采用典型的 MVC 模式，为了更好的将网页设计与 php 逻辑实现分开，我们采用了 Smarty 模版引擎技术。将美工页面(不包含任何的 php 代码)指定为模版文件，并将这个模版文件中有活动的内容，如数据库输出-用户交互等部分，定义成使用特殊“定界符”包含的“变量”，然后放在模版文件中相应的位置。当用户浏览时，由 php 脚本程序打开该模版文件，并将模版文件中定义的变量进行替换。对相应数据库表操作的实现封装在类中，由 php 逻辑实现部分调用相应对象的方法获取数据。

发布模块发布数据和取消发布数据采用类方法将查询数据库获取的数据转换成对应的点播 XML 数据，通过 php fsocketopen post 到需要发布到前端发布平台。

### 5.2 前端发布平台

前端发布平台通过 POST 数组取到后台发布过来的数据，根据发布过来的 XML 数据动态添加、删除、修改本地维护的专辑、音频等信息。一方面处理相应的数据，维护发布的专辑音频资源等信息；另一方面

前端发布平台需要实现终端的 http 强求，并给其返回合适的 json 数据。

### 5.3 Sencha Touch 终端

Sencha Touch 应用程序是一个由 View、Controller、Model、Store、Profile 组成的集合，外加一些额外指定的元数据。

终端是基于 Sencha Touch 框架搭建，采用 MVC 模式，通过在 View 中定义界面布局，总的分为首页、直播、点播、互动四部分。其中每个 view 中定义的组件一般通过指定 store 数据源对象来实现的。比如首页中点播处放的是个 list，通过指定 list 的 store，在 controller 中为该 store 动态加载数据。每个 store 的定义都绑定一个 model 和 proxy，model 属性指明返回数据的模型，主要是告诉程序返回数据是什么结构；proxy 属性指明返回数据的获取方式。我们根据网络广播的专辑、音频资源、电台信息等构建 model，store。

事件处理，人机交互和 UI 组件之间的切换，需要事件来驱动或触发，因此各 UI 组件都支持一系列特定的事件及其处理方法，定义在 Controller 里面。在 Controller 里面我们一系列事件的处理，以及数据的获取，通过调用前端发布平台的脚本返回对应的 json 数据并给对应的 store，在通过 view 显示出来。这样就实现了对发布的信息的动态展示。

## 6 测试

从后台管理平台添加并发布一个节目如“上下五千年”，发布数据格式如下：

```

<?xml version="1.0" encoding="utf-8"?>
<data>
<category>精品</category>
<Name>上下五千年</Name>
<Author>林汉达</Author>
<rtspAddr>rtsp://192.168.139.63/shangxiawuqianni
an.sdp</rtspAddr>
.....
</data>
  
```

前端发布平台接收并动态处理发布的数据，在终端访问时返回相应的 json 数据，终端接收并显示如图 4。

现在通过终端浏览并收听广播节目时，用户访问的流的地址是发布服务器的流地址，成功完成收听，由此证明了发布服务器实现了从源服务器的流媒体 Relay

功能. 测试环节从后台管理上传发布资源到终端资源的访问, 测试了网络广播流媒体发布系统的功能.



图4 Sencha Touch 业务终端截图

## 7 结论

本文针对网络广播设计开发了一个流媒体发布系统, 对媒体流的 Relay 和流媒体的发布两部分的设计与实现做了详细的描述. 最后通过测试验证了发布流程的可行性以及发布服务器的功能.

对流媒体发布系统来说, 需要进一步解决的问题: 发布服务器动态的请求源的问题、处理异构终端、不同的网络特性, 流格式等问题.

(上接第5页)

- Merge: Simplified Relational Data Processing on Large Clusters. Proc. of the 2007 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2007: 1029–1040.
- 20 Nykiel T, Potamias M, Mishra C, Kollis G. MRShare: Sharing Across Multiple Queries in Map-Reduce. Proc. of the VLDB Endowment, 2010,3(1-2):494–505.
- 21 Choi H, Son J, Cho YH, Sung MK. SPIDER: A System for Scalable, Parallel/Distributed Evaluation of large-scale RDF Data. Proc. of the 18th ACM conference on Information and Knowledge Management. New York: ACM, 2009: 2087–2088.

## 参考文献

- 唐静. 网络广播的优势及前瞻. 管理观察, 2009, (6): 203–204.
- 计叶. 基于 Darwin Streaming Server 快速流媒体中继实现. 中国科技论文在线.
- 夏斐. 实时视频监控系统中流媒体中继服务器的设计与实现[学位论文]. 北京: 北京邮电大学, 2011.
- 黄拔峰, 钟明, 杨传钧, 等. Darwin streaming Server 的研究与应用. 计算机工程, 2004, 30(19): 134–135.
- 杨碧华. 基于 Darwin 流媒体服务器的视频综合集成研究与应用[学位论文]. 广州: 暨南大学, 2011.
- senchaTouch2.0. [http://docs.sencha.com/touch/2-0/#\[S\]](http://docs.sencha.com/touch/2-0/#[S]).
- 袁进坤. 流媒体服务器中线程与缓存结构分析与研究[学位论文]. 南京: 南京邮电学院, 2004.
- Apple Inc. QuickTime streaming server modules programming guide. [http://images.apple.com/quicktime/pdf/QTSS\\_Modules.pdf](http://images.apple.com/quicktime/pdf/QTSS_Modules.pdf), 2005.
- Apple Computer Inc. QTSS Modules Programming Guide, 2006.
- 曲柳莺. 流媒体传输协议的研究[学位论文]. 成都: 电子科技大学, 2005.
- 蒋铮. 基于达尔文流媒体服务器 TS 流推送模块的设计与实现[学位论文]. 成都: 电子科技大学, 2011.
- 张伟, 都志辉, 李三立. 达尔文流媒体服务器用户认证扩展与应用. 小型微型计算机系统, 2010, 31(4): 706–710.