

基于 JMF 的跨平台视频点播系统的设计与实现^{*}

周贵芳, 方贵明

(中国科学院 软件研究所 多媒体中心, 北京 100080)

摘 要: 利用 Java 媒体框架(JMF)开发的视频点播系统具有开发简单、跨平台、提供 QoS 保证和交互性好等优点,具有广泛的应用前景。介绍了 JMF 应用编程接口的特点和结构,阐述了基于 JMF 开发的跨平台的视频点播系统的整体结构设计,并详细介绍了基于 RTP 的实时媒体传输、基于 RTCP 的拥塞控制、完整 VCR 操作和穿越 NAT 等关键技术的具体实现。

关键词: Java 媒体框架; 视频点播; 实时传输协议

中图分类号: TP393 文献标志码: A 文章编号: 1001-3695(2007)04-0235-04

Design and Implementation of Platform Independent Video on Demand System Based on JMF

ZHOU Gui-fang, FANG Gui-ming

(Multimedia Center, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Java Media Framework(JMF) is an API exploited by Sun Microsystems for simplifying the development of media streaming applications. The video on demand system based on JMF is easily developed, platform independent, QoS guaranteed and has a good interaction. It has the widespread application prospect. This paper introduced the characteristics and architecture of JMF. It presented the architecture of platform independent video on demand system based on JMF. And it discussed some key problems, including real-time media streaming based on RTP, congestion control based on RTCP, full VCR control operations and NAT traversing in detail.

Key words: JMF(Java Media framework); video on demand; RTP(real-time transport protocol)

0 引言

随着网络带宽的拓宽和计算机技术的发展,以视频点播为代表的实时流媒体业务得到了飞速的发展。视频点播业务以丰富的内容、自主的点播形式等特点受到了广大网络用户的欢迎。但是,现今市场上的视频点播系统还有许多缺点:不支持跨平台应用部署。通常的系统服务器端为 Linux,客户端为 Windows,如需要改变其部署操作系统,则需要重新开发。直接采用 UDP 或 TCP 进行流媒体数据的传输,没有一定的 QoS 保证。VCR 操作简单,不支持穿越 NAT 等。

Java 语言具有跨平台可移植性,在网络上应用广泛而且灵活。Sun 公司推出的 Java 语言的媒体软件开发包 JMF(Java Media Framework, Java 媒体框架^[1])使得流媒体应用开发变得简单。基于 JMF 的优点,笔者设计开发了一个跨平台的视频点播系统,这个系统具有开发简单、成本低;跨平台、可移植性好;采用实时传输协议^[2](RTP/RTCP)进行流媒体数据传输,服务质量(QoS)有保证;提供完整的 VCR 控制操作,交互性好;支持穿越 NAT 等优点。

1 JMF 介绍

JMF 是一个把音频、视频及其他基于时间的媒体结合到

Java 程序和 Applet 中的应用程序接口。它使得 Java 程序可以支持以下功能:捕捉和存储媒体数据,控制回放时执行的处理类型,进行媒体数据流的个性化处理。而且,最新版本的 JMF 还支持用户利用插件应用程序接口(Plug-in API)自定义和扩展 JMF 的功能。JMF 不仅支持本地媒体文件、媒体设备的采集、处理、存储、播放等功能,还支持基于网络的压缩媒体流的同步、控制、处理和播放。所谓媒体流,指的是可以边下载边播放的媒体传输方式。

JMF 包括 JMF API 和 JMF RTP API 两个部分。前者的主要功能是捕捉、处理、存储和播放媒体;后者主要是在网络上传输和接收媒体流。

1.1 JMF API 的特点与结构

JMF API 支持媒体捕捉并能够满足应用程序开发人员对媒体处理和播放进行控制的需要。JMF API 还提供了一个可以直接访问处理底层媒体数据的插件架构,使得 JMF 可以很容易地被扩展。它的特点有:易于编程;支持捕捉媒体数据;使 Java 可以开发媒体流传输和媒体会议等应用;使得高级开发人员和供应商能够在已存在的 API 基础上实现自定义解决方案,并能够方便地在已存在的框架中集成新的特征;提供对原始的媒体数据的存取等。通过利用 Java 平台的优势,JMF 保持了“Write once, run anywhere”的承诺,为访问底层的媒体

收稿日期: 2006-02-23; 修返日期: 2006-04-10 基金项目: 国家自然科学基金资助项目(60272078)

作者简介:周贵芳(1982-)男,江西铜鼓人,硕士,主要研究方向为多媒体网络通信(guifang03@iso.cn);方贵明(1967-),男,研究员,主要研究方向为社区宽带接入网和多媒体传输技术。

框架提供了一个通用的跨平台 Java API。

JMF 为基于时间的媒体数据的获取、处理和传送提供了一个统一的框架和消息协议,它可以支持大部分标准的媒体类型。JMF API 的结构主要由以下几部分组成:捕捉设备、数据源、播放器和处理器等。数据源类似于磁带,封装了捕捉设备或者媒体文件获取的媒体流;播放器类似于录像机,将数据源中的数据播放出来,输出到显示器、声卡等设备;而处理器类似于转录机,将数据源中的数据经过某些处理后得到另一个数据源。以上的这些部分都是以接口的形式出现,JMF 通过使用被称为 Manager(管理器)的中间对象来创建适应不同应用要求的接口。

数据源和播放器是 JMF 中用来管理捕捉、表现和处理基于时间的媒体的高层 API。JMF 还提供用于集成用户化的处理组件和扩展应用的低层 API,用于扩展编解码器、复用/解复用器等。

1.2 JMF RTP API 的特点与结构

JMF RTP API 提供了对实时传输协议 RTP 的支持。RTP 是一种应用型的传输层协议,用于 Internet 上针对多媒体数据流的传输。RTP 被定义为在一对一或一对多的传输情况下工作,其目的是提供时间信息并实现流同步。RTP 协议位于 UDP 之上,本身并不能为按顺序传送数据包提供可靠的传送机制,也不提供流量控制或拥塞控制,这些功能由 RTCP 来完成。RTP 和 RTCP 配合使用,它们能以有效的反馈和最小的开销使传输效率最佳化,因而特别适合传送网上的实时数据。JMF 对 RTP 的支持使用户开发基于流媒体实时传输的应用,如视频点播、视频会议、在线直播等,变得更为简单。

JMF 使用 SessionManager(会话管理器)来建立和管理用于传输实时流媒体的 RTP 会话。会话管理器跟踪会话的参与者和被传输的流,从本地参与者的角度维护会话状态。将 JMF 框架和 RTP 结合就形成了如图 1 所示的 JMF RTP 总体结构。

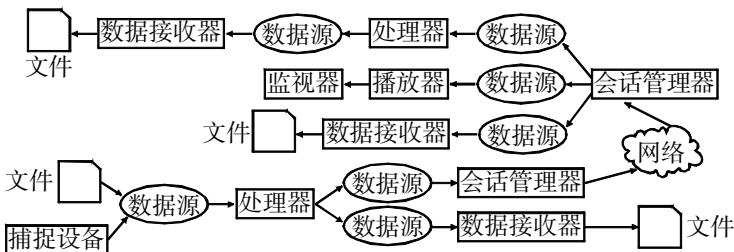


图 1 JMF RTP 总体结构

2 系统结构设计

系统可分为四个部分,即 Web 服务器、流媒体服务器、IP 网络和客户机。整个视频点播系统的结构如图 2 所示。各个部分的功能如下:

(1) 控制服务器。它由节目信息管理模块、用户信息管理模块和认证计费模块三个模块组成,负责系统数据传输以外的其他控制功能,主要功能都是基于 Web 服务器实现。节目信息管理模块负责节目信息的配置,并将配置好的节目信息发布到 Web 上,供用户点播;用户信息管理模块负责用户信息的维护;认证计费模块负责用户的注册、登录、点播记录,并根据点播情况和计费策略生成点播费用清单。

(2) 视频服务器。它由 RTSP 服务器和数据泵两部分组成,负责点播连接的建立、数据的传输及控制,主要功能都是基

于 Java 和 JMF 实现。RTSP 服务器负责接收处理各种 RTSP 请求,实现点播连接的建立、释放和各种 VCR 控制操作;数据泵负责流媒体数据传输。

(3) IP 网络。它既可以是宽带局域网,也可以是国际互联网。为支持流媒体的实时传输,IP 网络需要保证足够的带宽。

(4) 客户机。它由浏览器、数据接收器和 RTSP 请求发生器三部分组成,负责流媒体数据的接收和回放、用户与服务器的交互,主要是基于 Java Web 技术和 JMF 流媒体接收、播放技术实现。浏览器负责用户与控制服务器交互;数据接收器负责流媒体数据的接收、缓冲及回放,还要实现基于 RTCP 的拥塞控制信息反馈;RTSP 请求发生器负责与 RTSP 服务器交互,实现点播请求的发送、VCR 控制信令的发送等。

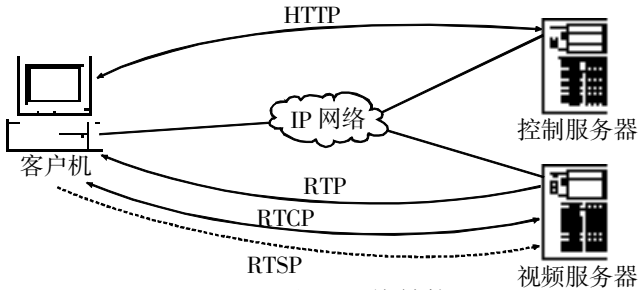


图 2 系统结构

点播进行过程可以描述如下:用户通过客户机的浏览器登录系统,选择节目并通过 RTSP 请求发生器发送点播请求。RTSP 服务器接收处理点播请求,选择流媒体服务器,定位媒体文件。数据泵从磁盘阵列上的媒体文件或者音视频采集设备等多种媒体源获取媒体流,经过特定编码处理后建立 RTP 会话,以 RTP/RTCP 将媒体数据传输给客户机。客户机从网络上接收 RTP 数据,边接收边播放,同时不停地发送 RTCP 反馈信息。在点播的过程中,用户可以通过 RTSP 请求发生器来进行各种 VCR 控制操作。

系统的实现和运行表明,采用控制信息和数据处理的分开数据传输和传输控制的分开,能够降低系统实现的复杂度,采用模块化的实现方法可以提高实现效率,简化系统的调试和维护。客户机和服务器之间采用标准的协议(RTSP、RTP/RTCP、HTTP)进行交互,降低了两者的耦合程度,有利于对某一方面进行改进的时候不会对另一方造成影响。

3 关键技术及实现

视频点播系统的技术指标包括音视频的回放质量和连续性、传输的实时性、友好灵活的交互性、对各种网络环境和网络变化情况的自适应性等。要实现这些指标,基于 RTP 的实时媒体流传输^[3]、基于 RTCP 的拥塞控制^[4]、完整 VCR 控制、穿越 NAT^[6]等技术是关键的技术。

3.1 基于 RTP 的实时流媒体传输

基于 JMF 对媒体流和实时传输协议 RTP 的支持,采用 JMF 可以很方便地实现基于 RTP/RTCP 的实时媒体流传输。传输的建立过程可以分为以下两部分。

(1) 创建并配置处理器对象,产生数据源

处理器(Processor)对象是用来设置和控制从媒体源读取媒体数据的。媒体源可以是媒体文件也可以是音视频采集设备,这里以媒体文件为例。创建处理器对象的方法是:新建以媒体文件路径为参数的 MediaLocator 对象,以此对象为参数调

用 Manager(管理器) 对象的 createProcessor 方法。

配置处理器对象首先要对处理器对象的通道(Track) 设置参数。方法是: 调用处理器对象的 getTrackControls 方法, 得到通道的 TrackControl 对象, 然后调用 TrackControl 对象的 setFormat 方法设置处理后的媒体格式(即适合于 RTP 传输的媒体编码格式)。对于音频, JMF 提供五种编码格式, 即 ULAW/RTP、G723/RTP、GSM/RTP、DVI/RTP 和 MPEGAudio/RTP; 对于视频, JMF 提供四种编码格式, 即 JPEG/RTP、MPEG/RTP、H261/RTP 和 H263/RTP。采用哪种编码格式依赖于媒体文件存储时的压缩编码格式, 如以 XVID、MP3 压缩的 MPEG4 媒体文件可以采用 H263/RTP、MPEGAudio/RTP 编码格式进行传输。

配置处理器对象还要设置处理器对象的数据输出格式, 方法是调用处理器对象的 setContentDescriptor 设置。可设置的数据输出格式有 CONTENT_UNKNOWN、RAW、MIXED、RAW_RTP; 一般采用默认值 RAW 即可。例如编码格式为 JPEG/RTP, 还可以通过调用 JPEG 编码格式的 QualityControl 对象的 setQuality(float newQuality) 方法设置图像质量, 一般设置为 0.5。处理器对象配置完成后就可以调用 getDataOutput 方法获取数据源, 以供下一步使用。

(2) 创建传输器, 进行 RTP 数据流传输

JMF 中传输 RTP 数据流有两种方法: 使用数据接收器(DataSink) 进行数据流传输; 使用会话管理器(SessionManager) 创建输出数据流并控制传输的进行。

使用数据接收器进行传输是比较简单的方法, 只需要创建一个数据接收器对象, 然后调用此对象的 open()、start() 方法来开启传输, 调用 stop() 方法来停止传输。但是, 使用数据接收器进行传输只能传输一个数据流, 而且无法监控会话状态。在本系统中, 一般传输的媒体文件均包括音频、视频两个数据流, 而且利用 RTCP 进行传输拥塞控制时需要监控会话状态, 所以无法使用数据接收器来进行传输。这时候就需要使用会话管理器创建多个输出数据流来进行传输。

使用会话管理器传输音视频 RTP 数据流的方法如下:

- (1) 创建一个会话管理器对象(实例化一个 RTPSessionMgr 对象);
- (2) 调用 initSession 方法初始化会话管理器;
- (3) 调用 startSession 方法开启会话管理器;
- (4) 调用 createSendStream 方法创建输出流对象;
- (5) 调用输出流对象的 start() 方法开始 RTP 数据流的传输。

3.2 基于 RTCP 的拥塞控制

Internet 是不能提供带宽保证的网络, 很容易发生拥塞。而且 Internet 接入用户的网络情况带宽都不太一样, 发生拥塞的情况也不一样。如果以单一的传输速率进行传输, 对于连接速率低于设定的传输速率的用户, 将会出现频繁缓冲的现象; 而对于连接速率高于传输速率的用户来说, 浪费了部分带宽, 接收的效果并没有达到最好。为了实现 Internet 视频点播系统的应用级 QoS 保证, 可以根据 RTCP 反馈的传输统计信息, 动态地对 RTP 传输速率进行调整, 以适应网络带宽的变化, 提高音视频的传输质量。

判断网络拥塞有不同的策略, 这里主要根据某段时间内的丢包率的大小作为标准来判断。自适应流媒体传输控制策略

分为三个步骤:

- (1) RTCP 分析。根据接收包 RR 的反馈信息得出包丢失率等参数。
- (2) 网络状况分析。根据包丢失率的大小分析出网络拥塞状况。
- (3) 带宽调整。根据网络拥塞状况调整流媒体数据的发送速率。

本系统采用 AIMD(加性增、乘性减) 算法对传输速率进行动态调整。具体算法如下所示:

```
if ( p > Pc )
    R = max{ R * i, MinR}
else if ( p < Pu )
    R = min{ R + AIR, MaxR}
else
    R = R;
```

其中, p 是丢包率; R 是传输速率; P_c 是网络拥塞临界值; P_u 是网络轻载临界值; $MaxR$ 是允许的最大发送速率; $MinR$ 是允许的最小发送速率; AIR 是速率线性增量; i 是乘性减小因子。输入参数丢包率 p 既可以是原始值, 也可以是平滑值, 使用平滑值可以减少抖动。本系统使用一个低通滤波器对丢包率 p 进行平滑处理: $p = (1 - a)p + ab$ 。其中 b 为 RR 包中的丢包率; a 是滤波调整参数, $0 < a < 1$ 。

丢包率 p 的计算方法如下: 调用输出流对象的 getSenderReport() 方法得到接收端报告, 调用 getSenderPacketCount() 方法得到接收到的包数 pLost; 调用输出流对象的 getSourceTransmissionStats() 得到传输状态, 再调用 getPDUTransmitted() 方法得到已发送的包数 pSent。因此, $p = (pSent - pLost) / pSent$ 。速率调整可以通过调用输出流对象的 setBitRate 方法实现。

3.3 完整 VCR 控制

基于用户对视频流控制的程度可以将 VOD 系统分为两种: TVOD(True VOD) 系统中, 用户可以完全控制视频节目的播放, 进行各种 VCR 控制操作, 如暂停、播放、快进、快退以及随机定位节目播放位置; NVOD(Near VOD) 系统中, 用户只能对节目播放有部分的控制能力。本文实现的是支持完整 VCR 控制操作的 TVOD。

在某些直接使用 TCP、UDP 或 HTTP 传输媒体文件数据的 TVOD 中, 只能通过改变 VOD 服务程序数据发送状态和发送内容来实现简单的 VCR 操作, 如暂停、播放、拖动等。因为不对媒体文件编码格式进行处理, 所以无法实现快进、快退等更深层次的 VCR 操作。JMF 中流媒体服务程序分为处理器和传输器两个部分, 在处理器部分对媒体文件的编码格式、编码内容进行了处理, 只需要对处理器作相应的控制, 就可以很方便地实现快进、快退等深层次的 VCR 操作。

具体的各种 VCR 控制操作的软件实现如下:

- (1) 暂停、播放。调用处理器对象(Processor) 的 stop()、start() 方法实现。
 - (2) 随机定位节目播放位置。以新播放位置构造的时间对象(Time) 为参数, 调用处理器对象(Processor) 的 setMediaTime() 方法实现。
 - (3) 快进、快退。调用处理器对象(Processor) 的 setRate() 方法实现, 快进速率参数为正, 快退为负。
- 需要注意的是, (2) 和(3) 中调用 set 方法之前需要确保处

理器对象为 stopped 状态。

3.4 穿越 NAT

Internet 是一种复杂的网络环境,当前普遍运行的互联网版本 IPv4 的一个显著的缺点就是 IP 地址不足。许多互联网用户不得不使用 NAT(Network Address Translation,网络地址转换) 来解决 IP 地址不足的问题。对于这些主机,外部的主机不能直接访问到它们。目前,许多的流媒体传输软件还不能解决 NAT 内部和外部、两个不同的 NAT 内部的用户之间的传输问题。因此,穿越 NAT 进行实时流媒体的传输是视频点播软件迫切需要解决的一个问题。

穿越 NAT 技术的原理即为俗称的“打洞”技术。NAT 内外用户的呼叫方向受到限制,内部用户可以呼叫外部用户,外部用户不能直接呼叫内部用户。虽然呼叫的方向受到限制,但是数据传输的方向与呼叫的方向无关。也就是说,一旦呼叫得到应答,连接建立,数据传输可以是双向的。因此,由内部用户先呼叫外部用户,外部用户应答,从而建立连接,然后两者利用这个连接收发数据,实现 NAT 的穿越。

许多常用的 P2P 通信软件只实现了文本数据传输的 NAT 穿越,而没有实现音视频数据传输的 NAT 穿越。这是因为音视频数据传输的通道需要另外建立,而不能利用已经“打好洞”的通道进行数据传输。JMF 解决了这个问题,JMF 允许用户使用用户化的、处理过的 TCP 或者 UDP 通道替代原有的通道进行数据传输,实现音视频数据传输的 NAT 穿越。

具体的 JMF 穿越 NAT 实现如下:

(1) 客户机开启所需 UDP 通道(DatagramSocket),针对每一个通道,向服务器发送“打洞包”。

(2) 服务器开启所需 UDP 通道并接收“打洞包”,记录下每个通道所收到的“打洞包”的来源地址和端口,即媒体数据发送的目的地址和端口。

(3) 客户机和服务器分别使用上两个步骤“打洞”后的 UDP 通道构造适配器类 RTPClientSocket Adapter 和 RTPServerSocketAdapter 的对象。这两个适配器类实现了 RTPConnector 接口,利用“打洞”后的 UDP 通道发送和接收媒体数据。

(4) 客户机和服务器分别以步骤(3)中的两个适配器对象为参数,调用 RTPManager 类的 initialize (RTPConnector conn) 方法初始化会话管理器,然后建立 RTP 会话进行媒体传输。

本系统中的流媒体服务器一般位于公网中,即 NAT 外部,所以只考虑了 NAT 内部和外部之间穿越 NAT 的情况。对于双方都位于 NAT 内部的流媒体传输情况可以根据双向打洞原理,依靠一个中转服务器同理实现。

4 结束语

本文介绍了 Java 媒体框架应用编程接口 JMF 的结构特点,阐述了利用 JMF 开发的跨平台视频点播系统的系统结构设计和关键技术,详细介绍了相关关键技术的具体软件实现。所开发的视频点播系统具有跨平台、提供一定的 QoS 保证、实现了完整的 VCR 操作、穿越 NAT 等优点。随着宽带的普及和 JMF 技术的发展,基于 JMF 的视频点播系统将具有非常好的应用前景。

参考文献:

[1] ava media framework[EB/OL] . (2006- 02- 23) . <http://java.sun.com/products/java-media/jmf/index.jsp>.

[2] SCHULARINNE H, CASNER S, FREDERICK R. RFC 1889 RTP A transport protocol for real-time applications[S] . [S. l.] : [s. n.], 1996.

[3] 赵新慧, 李文超. 基于 JMF 的 Internet 可视通话软件的设计与实现[J] . 计算机应用研究, 2005, 22(4) : 93-95.

[4] FORTINO G, FURFARO A, GUERRI C, *et al.* A Java-based adaptive media streaming on-demand platform: proc. of SCS Euromedia [C] . Modena: [s. n.], 2002: 15-17.

[5] 傅 骞, 曲朝霞. 基于 JMF 的内外网多媒体数据传输技术[J] . 计算机工程, 2004, 30(11) : 141-142, 167.

[6] ZHOU S, KING P. A simple platform independent video on demand application: International Conference on Information Technologies and Applications[C] . Bathurst: [s. n.], 2002: 215.

[7] 杨际祥, 罗代东, 高远. 基于 MPEG-4 的视频点播系统的设计与实现[J] . 计算机应用与软件, 2004, 21(10) : 37-38, 128.

(上接第 183 页)

但是本文方法中的区域划分和权重设置完全是根据人的经验确定的,并且是针对英文字符这样的特定对象设置的。为了适应各种不同的目标对象,今后的研究将集中在如何自动确定最佳的划分区域,以及如何获取最佳的权重,使得匹配效果更好。

参考文献:

[1] 丁晓青. 汉字识别研究的回顾[J] . 电子学报, 2002, 30(9) : 1364-1368.

[2] HUTTENLOCHER D, KLANDERMAN G, RUCKLIDGE W. Comparing images using the Hausdorff distance[J] . **IEEE Trans. on Pattern Analysis and Machine Intelligence**, 1993, 15(9) : 850-863.

[3] EFFICIENTLY R W J. Locating objects using the Hausdorff distance [J] . **International Journal of Computer Vision**, 1997, 24(3) : 251-270.

[4] 彭晓明, 丁明跃, 周成平, 等. 一种利用 Hausdorff 距离的高效目标

搜索算法[J] . 中国图像图形学报, 2004, 9(1) : 23-28.

[5] KWON O K, SIM D G, PARK R H. Robust hausdorff distance matching algorithms using pyramidal structures[J] . **Pattern Recognition**, 2001, 34(7) : 2005-2013.

[6] WU Xuejing, SHI Pengfei. Unconstrained handwritten numeral recognition using Hausdorff distance and multi-layer neural network classifier: proceedings of the 5th International Conference on Document Analysis and Recognition[C] . Bangalore: [s. n.], 1999: 249-252.

[7] DUBUISSON M P, JAIN A K. A modified Hausdorff distance for object matching: proceedings of International Conference on Pattern Recognition[C] . Jerusalem: [s. n.], 1994: 566-568.

[8] LU Yue, TAN Chewlim, HUANG Weihua, *et al.* An approach to word image matching based on weighted Hausdorff distance: proceedings of the 6th International Conference on Document Analysis and Recognition[C] . Seattle: [s. n.], 2001: 10-13.

[9] LU Yue, TAN Chewlim. Chinese word searching in imaged documents[J] . **International Journal of Pattern Recognition and Artificial Intelligence**, 2004, 18(2) : 229-246.