

# 流媒体直播(时移)系统的设计

徐良基 叶德建

(复旦大学软件学院宽带网络与互动多媒体实验室, 上海 200433)

**摘要** 随着多媒体技术的发展和宽带网络的普及,流媒体技术得到越来越广泛的应用,其中,尽管流媒体直播技术定位于提供实时流服务,然而,国内外现有直播系统均未能很好解决直播系统的实时性与用户的个性化需求之间的矛盾。针对这个问题,首先设计了一个具有完全自主知识产权的支持时移功能的流媒体直播系统,并创新地采用一种实现时移的时间计算方法。目前该系统已成功应用于上海网络电视试点的大宁模式中。

**关键词** 流媒体 直播 时移 网络电视

**中图分类号**: TN919.85 **文献标识码**: A **文章编号**: 1006-8961(2006)11-1516-04

## The Design of Live(Time-shift) Streaming System

XU Liang-ji, YE De-jian

(Multimedia and Networking Lab, Software School, Fudan University, Shanghai 200433)

**Abstract** Due to the real-time characteristic, live streaming applications attract more and more attentions. Most existing live streaming applications provide streaming clients with the same data at the same time, which do not match with each user's requirement. In this paper, a novel software architecture is proposed which can be used to implement a time-shifting live streaming system. Besides, this paper also proposes a novel algorithm to calculate the play time for the time-shifting system. At last, a system based on the proposed architecture and algorithm has been developed and deployed successfully in Shanghai Daning IPTV (internet-protocol television) systems.

**Keywords** video streaming, live streaming, time-shifting, internet-protocol television (IPTV)

## 1 引言

流媒体技术以流的形式在网络上传送音频和视频数据流,由于其具有高效和有利于版权保护等特点,因而得到越来越多的关注。典型的流媒体应用包括视频点播和视频直播等<sup>[1-3]</sup>。

直播系统的实时性虽可极大地满足了用户欣赏实时内容的需求,然而,实时性与用户的需求有时又是矛盾的。实时性要求系统中所有播放器在同一时刻播放的内容是大致相同的,然而,用户可能因故暂时离开而希望暂停当前正在直播的内容,也会可能因为内容的精彩而希望重温已完成直播的内容,还

可能因为喜好而要快速浏览已完成直播的内容<sup>[4]</sup>,也可能在这些操作完成后要随时切换到当前正在直播的内容,而这些都与直播系统的实时性是矛盾的。虽然现有点播系统支持这些功能,但点播系统只支持预先编码和存储的数据,并不提供实时性的支持。

在这里,可把直播系统支持的暂停、拖动、快进和快退等功能统称为时移功能,将支持时移功能的直播系统称为直播(时移)系统,而暂停和拖动等操作则称为时移操作。

流媒体(特别是IPTV-internet protocol television)直播系统的实时性与用户的个性化需求之间的矛盾已阻碍了流媒体直播应用的发展<sup>[5]</sup>。更为重要的是,目前国内和国际上大部分直播系统均未能很好

**基金项目**:国家自然科学基金项目(60503044)

**收稿日期**:2006-06-28;**改回日期**:2006-08-03

**第一作者简介**:徐良基(1981~),男,2006年毕业于上海复旦大学软件学院,获软件与理论专业工学硕士学位。主要研究方向为宽带网络与互动多媒体。E-mail:032053009@fudan.edu.cn

地支持能解决该矛盾的时移功能。

## 2 系统架构

针对现有直播系统中普遍存在的这个问题,笔者设计了一个支持时移功能的流媒体直播服务器:Clear 直播服务器。该服务器采用一种创新的时间计算方法来实现实时时移功能。

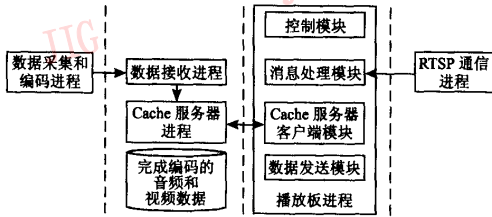


图 1 Clear 直播服务器架构图

Fig.1 Architecture of Clear live server

Clear 直播服务器架构如图 1 所示。其中采集和编码进程用于多媒体数据的实时采集和编码;数据接收进程用于从数据采集和编码进程中接收数据;Cache 服务器进程用于管理所有频道的数据,并根据请求向播放板进程提交数据;播放板进程则根据 RTSP(real-time streaming protocol)通信进程的指示向播放器发送数据;RTSP 通信进程用于处理播放器的 RTSP 请求,并将处理结果以控制消息方式发送给播放板进程。图中进程间的连接线表示进程之间存在数据或控制信息的通信,箭头方向表示进程间通信数据的流向。

在图 1 所示架构中,服务器的计算能力、网络带宽和磁盘 I/O 能力容易成为系统的瓶颈,由于 RTSP 通信数据量小,因此,可将数据采集和编码进程和 RTSP 通信进程分别运行于独立的服务器中,其余 3 个进程运行于同一台服务器中,根据进程名称和功能分别将服务器命名为采集服务器、主控服务器和播放服务器。由于单台主控服务器可管理多台播放板服务器,因此,系统可通过灵活部署来实现所有用户的集中控制,以支持多种编码格式和多种客户端。同时,系统可充分利用多台播放服务器的带宽来提高整个系统的并发数。根据这种部署,在直播过程中,客户端首先从 Web 服务器获取直播服务器和频道相关信息,并向主控服务器提交频道请求,主控服务器处理请求后,先将控制信息发送给播放服务器,然后播放服务器据此向播放器发送数据(如图 2 所示)。

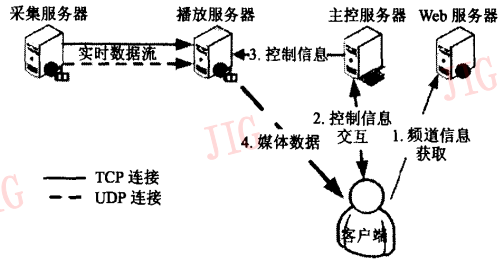


图 2 Clear 直播应用示意图

Fig.2 Illustration of Clear live server

## 3 时移的关键实现

在流媒体直播系统中,由于数据可通过实时采集和编码获得,因而数据的播放时间取决于数据的采集和编码时间。在本系统中,播放器根据相对时间请求数据,而服务器则根据相对播放时间和服务器绝对时间定位和发送数据。

记直播频道的最大时移时间为  $t_{MaxDuration}$ ,所有相对播放时间的值非负,且不大于  $t_{MaxDuration}$ 。系统约定最大时移时间对应于服务器当前的绝对时间,而相对播放时间均以该绝对时间为参照。相关播放时间的计算如下:

### (1) 正常播放

正常播放是指快进快退倍数  $K_{Scale}$  值为 1 的播放,此时会话的播放时间始终不变。

### (2) 暂停后重新播放

会话在绝对时间为  $t_{abPause}$  时暂停,此时的播放时间为  $t_{prePlay}$ ,并在绝对播放时间  $t_{abPlay}$  重新开始播放,此时的播放时间为  $t_{curPlay}$ 。

当  $t_{prePlay}$  不小于  $t_{abPlay}$  与  $t_{abPause}$  之差时,  $t_{curPlay} = t_{prePlay} - (t_{abPlay} - t_{abPause})$ ; 当  $t_{prePlay}$  小于  $t_{abPlay}$  与  $t_{abPause}$  之差时,则  $t_{curPlay} = 0$

### (3) 快进的播放

会话以快进或快退倍数  $K_{Scale}$  (大于 1 的整数,一般为 2、4、8、16、32 和 64) 从绝对快进起始时间  $t_{FFStart}$  起快进,此时的播放时间为  $t_{prePlay}$ ,并在绝对快进终止时间  $t_{FFEnd}$  转为  $K_{Scale}$  为 1 的正常播放,此时的播放时间为  $t_{curPlay}$ 。

当  $t_{prePlay}$  与  $(t_{FFEnd} - t_{FFStart}) \times (K_{Scale} - 1)$  之和大于最大时移时间时,则  $t_{curPlay} = t_{MaxDuration}$ , 否则  $t_{curPlay} = t_{prePlay} + (t_{FFEnd} - t_{FFStart}) \times (K_{Scale} - 1)$

(4) 快退的播放

会话以倍速  $K_{Scale}$  (小于 0 的整数, 一般为 -2、-4、-8、-16、-32 和 -64) 从绝对快进开始时间  $t_{FRStart}$  起快退, 此时的播放时间为  $t_{prePlay}$ , 并在绝对快进终止时间  $t_{FREnd}$  时结束快退, 转为  $K_{Scale}$  值为 1 的正常播放, 此时的播放时间为  $t_{curPlay}$ 。

当  $t_{prePlay}$  与  $(t_{FREnd} - t_{FRStart}) \times (K_{Scale} - 1)$  的和小于 0 时, 则  $t_{curPlay} = 0$ , 否则  $t_{curPlay} = t_{prePlay} + (t_{FREnd} - t_{FRStart}) \times (K_{Scale} - 1)$

(5) 相对播放时间与绝对时间的对应关系

相对播放时间非负, 且不大于最大时移时间。记当前系统的绝对播放时间为  $t_{abCurSys}$ , 与相对播放时间  $t_{curPlay}$  对应的绝对播放时间为  $t_{abCurPlay}$ , 则

$$t_{abCurPlay} = t_{abCurSys} - (t_{MaxDuration} - t_{curPlay})$$

由于播放器采用相对时间请求数据, 因此不需要与服务器保持精确的时间同步。服务器采用系统绝对时间来标记数据的播放时间。通过这种方法, 不仅服务器能方便定位和发送数据, 而且客户端也能方便地请求和播放数据, 因此整个系统可以支持长时间的时移。

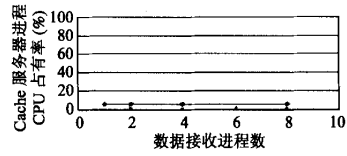
4 系统测试

在系统实现的基础上, 还进行了 Cache 服务器进程与数据接收进程关系、磁盘 I/O、数据接收进程性能和播放板并发性能等 4 项性能测试。测试用的视频流码率约为 4.5Mbps, 视频特性接近 CBR (constant bitrate)。

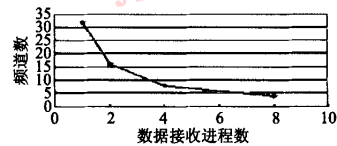
图 3 为 Cache 服务器进程与数据接收进程关系的测试结果。测试中保持 Cache 服务器进程承载的频道总数不变, 并依次将频道平均配置在不同数量的播放板进程上。测试结果表明, 在频道总数不变的情况下, 随着数据接收进程数的增加, Cache 服务器进程的 CPU 占有率基本不变, 即 Cache 服务器进程的 CPU 占有率不受数据接收进程数的影响。

图 4 为磁盘 I/O 性能测试结果。在测试中, 对测试中的播放器请求的不同频道, 或请求相同频道的不同播放时间, 以及同时在每组测试完成后重启 Cache 服务器进程进行了设置。测试结果表明, 系统所用存储系统最大 I/O 吞吐能力约为 35MB/s。

图 5 为数据接收进程的性能测试结果。在测试中, 运行单个 Cache 服务器进程, 并依次为数据接收进程配置不同的频道数。由于测试用的直播服务器



(a) Cache 服务器进程与数据进程关系测试



(b) 进程数与频道数关系

图 3 模块关系测试

Fig. 3 Module relationship testing

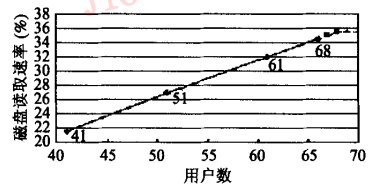


图 4 磁盘性能测试

Fig. 4 Performance of storage

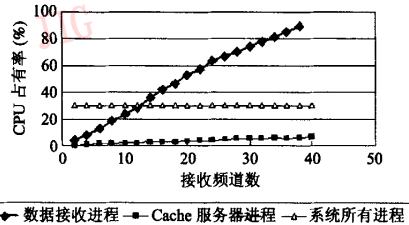


图 5 数据接收进程性能测试

Fig. 5 Performance of data receiving module

是采用双 Intel® Xeon™ 超线程 CPU, 其数据接收进程和 Cache 服务器进程的 CPU 占有率是针对单个 CPU 统计的, 而系统所有进程的 CPU 占有率则针对所有 CPU 资源统计, 所以图中存在系统所有进程 CPU 占有率低于数据接收进程 CPU 占有率的现象。测试结果表明, 随着频道数的增加, 数据接收进程的 CPU 占有率持续上升, 而 Cache 服务器进程及系统所有进程的 CPU 占有率则相对稳定。当频道数增加到 40 时, 数据接收进程的 CPU 占有率在 90% 上下波动, 此时所有频道的码率总和约为 180Mbps (以每个频道 4.5Mbps 计算), 远低于系统的可用网络

带宽。由于此时 Cache 服务器进程和存储设备 I/O 的负载远未达满负荷,而数据接收进程的功能则是从网络中接收组播的视频流数据,同时分析并对其中的关键帧进行标记,因此,数据的接收、分析和标记占用了大量的 CPU 资源,可见 CPU 是数据接收进程的瓶颈。

图 6 是直播播放板的并发性能测试结果。从图 6 可以看出,随着用户数的增加,尽管播放板进程占用的网络带宽随之增加,且播放板进程的 CPU 占有率很快升至接近 100%,但系统总的 CPU 占有率却波动很小。当用户数为 300 个时,播放板进程占用的网络带宽约为 170MB,远远超过单块千兆网卡所能提供的网络带宽。为了保证系统的实时性,对于播放时间为最大时移时间的会话,播放板进程会不停尝试读取最新的数据,这是由于播放板进程的 CPU 占有率很快升至接近 100% 的原因,因此,改进直播播放板的调度算法有助于提高系统支持的最大并发数。由于本组测试中 Cache 服务器进程的负载远未达满负荷,所以单个直播播放板进程最多能支持 300 个码率约为 4.5Mbps 的并发用户,可见网络带宽和 CPU 是直播播放板进程的瓶颈。

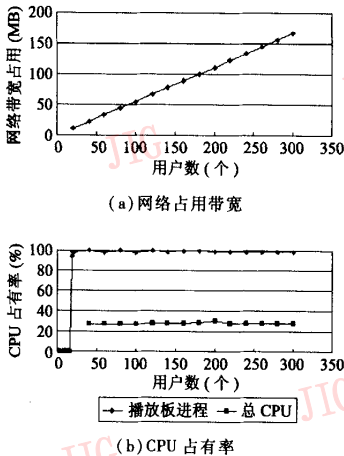


图 6 播放板并发性能测试

Fig. 6 Performance of data sending module

整个测试结果表明:(1)在支持时移功能的前提下,服务器不仅能充分利用系统的资源,而且整个系统的性能并未因支持时移功能而受到影响;(2)整个直播系统可通过灵活部署来支持多频道长时间时移和大并发用户;(3)磁盘 I/O 和网络带宽是系统的一个瓶颈,而改进 Cache 服务器中的数据管理算法对提高整个服务器的性能有重要意义。

## 5 结 论

IPTV 的发展推动了流媒体直播系统的应用,但仍存在一些问题。本文针对现有直播系统中普遍存在的实时性与用户的个性化需求之间的矛盾,设计了 Clear 直播服务器,系统创新地采用一种实现时移的时间计算方法。目前系统已成功应用于上海 IPTV 试点的大宁模式中。在已实现系统的基础上,如何解决现有直播系统中存在的缺乏对多种编码格式支持的问题将作为本文后续工作展开研究。

### 参考文献 (References)

- 1 Zhong Yu-zhuo, Xiang Zhe, Shen Hong. Video streaming and video server[M]. Beijing: Tsinghua University Press, 2003:1~82. [钟玉琢,向哲,沈洪.流媒体和视频服务器[M].北京:清华大学出版社,2003:1~82.]
- 2 Ye De-jian, Xu Liang-ji, Zhang Zuo. Characteristic-time-sequence-based BIBO stability analysis for video streaming buffer control problems[A]. In: ACM SIGCOMM(Special Interest Group on Data Communication) Asia Workshop[C], Beijing: China, 2005.
- 3 Ye De-jian, Wu Qiu-feng, Zhang Zuo. Receiver-buffer-driven approach to adaptive Internet video streaming[J]. Electronics Letters, 2002, 38(22): 1405~1406.
- 4 Dan Asit, Dias Daniel M, Mukherjee Rajat. Buffering and caching in large-scale video servers[A]. In: Comcon 1995[C], San Francisco, CA, USA, 1995: 217~224.
- 5 Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method[A]. In: Proceedings of the 6th International Conference on Advanced Communication Technology[C], Phoenix Park, Republic of Korea, 2004, 2: 971~975.