Linux 下 WMV 高性能 IPTV 流媒体服务器的设计与开发

孙澔峻 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 201203)

摘 要 流媒体点播技术定位于提供实时的文件点播服务,而 WMV 则是一种 IPTV 常用的文件格式。为了解决 Linux 或类 Unix 环境下对 WMV 文件格式的支持,设计了一个具有完全自主知识产权的支持 WMV 格式的 Linux 环境下的流媒体点播系统。该流媒体服务器采用创新的主从式架构设计,并设计了 RTSP 连接队列调度等算法,用来完善服务器的功能和提高服务器的性能。

关键词 流媒体 网络电视 WMV 高性能 Linux 平台
中图法分类号: TN919.85 **文献标识码**: A **文章编号**: 1006-8961(2007)10-1701-05

The Design and Implementation of An IPTV Video Streaming Server Supporting the WMV Format on Linux Platforms

SUN Hao-jun, YE De-jian
(*Multimedia and Networking Laboratory, Software School, Fudan University, Shanghai* 201203)

Abstract Video streaming applications provide on-demand video file playback services through the computer network. WMV is a common file format in IPTV applications. In order to support WMV format on Linux or Unix-like platforms novel master-slave mode architecture is proposed in this paper. Based on this architecture, we designed and implement a WMV video streaming system on Linux platforms. Inside the server, we embedded several novel algorithms such as the RTSP query schedule algorithm to improve the performance. The whole system is successfully deployed in the Shanghai's Archive Library.

Keywords video streaming, internet-protocol television (IPTV), windows media video (WMV), high performance, Linux platform

1 引 言

流媒体技术以流的形式在网络上传送音视频数据,由于具有高效和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直播等[1]。在流媒体服务器系统中,文件格式的组织是关键问题。考虑编解码效率、表现效果,由于 WMV(windows media video)压缩效果与画面质量跟 H.264 相当,因此 WMV 格式在 IPTV(internet-protocol television)流媒体应用中,也处于一个比较流行的地位,有一类 IPTV 就是采用 WMV 格式的。

然而,WMV 格式也有一定的限制。支持 WMV 格式的服务器目前主要运行在 Windows 平台,缺乏对 Linux 和其他类 Unix 操作系统的支持,可是 IPTV 的服务器却需要基于 Linux 或 Unix 平台。另外,对比这些操作系统,Windows 平台在安全性和稳定性上与 Linux 等操作系统相比有一定差距,难以满足流媒体服务器一天 24 小时一周 7 天的工作模式;而且由于 Windows 操作系统庞大,相对于不需图形界

基金项目:国家自然科学基金项目(60503044);上海市自然科学基金项目(05ZR14018)
收稿日期:2007-05-30;改回日期:2007-07-01
第一作者简介:孙澔峻(1982~),男。复旦大学软件学院研究生。研究方向为宽带网络与互动多媒体。E-mail: 0030006@fudan.edu.cn

面的 Linux 操作系统来说,要占用更多的系统资源,从而导致流媒体系统所能支持的并发数比不上同等硬件条件下的 Linux 流媒体服务器。

综合安全性、稳定性和硬件成本等因素,Linux 等平台优于 Windows 平台。因此,笔者致力于设计开发一种基于 Linux 或类 Unix 环境的支持 WMV 格式的自主 IPTV 流媒体服务器。该实现的服务器,在同等硬件条件下,比 Windows Media Server 并发用户数性能提升 30% 左右。因此实现支持 WMV 格式的流媒体服务器,对 Linux 下支持其他文件格式和多格式流媒体服务器的开发有参考价值。

2 问题描述

在笔者的设计方案中,WMV 作为一种独立的媒体文件,它的组织形式与服务器支持文件的组织形式如何匹配,是一个需要考虑的问题。

流媒体服务器一个重要的性能指标就是并发用户数。在同等硬件条件下,服务器能支持的用户数越多,相对的每一个用户的成本就越小,从而使得整个服务器更有实际应用的价值。服务器性能和规模的扩展性也是开发过程中一个重要的需要考虑的因素,要做到以尽量小的部署代价来获得性能的提升和扩展。

3 系统架构和关键点设计

3.1 架构及说明

笔者通过观察认为,软件流媒体服务器的总体功能主要分为相应协议的实现、任务调度功能和输入/输出(I/O)能力。而现有的服务器由于其中一些功能是由操作系统实现[2]的,为此,需将服务器的主要功能进行抽象,并由上层软件实现,以便达到与操作系统无关。

在将流媒体服务器各项功能分解之后,就可以根据各块协议与功能的具体特点来设计算法,达到专用化,以提高整个服务器的性能。

基于此,笔者设计了一个新颖的 IPTV 流媒体点播服务器——Clear 点播服务器。

Clear 点播服务器架构如图 1 所示。该服务器除了数据库和磁盘外,主要由主控服务器、RTSP(real-time streaming protocol)连接控制进程和播放板 3 个进程组成。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。

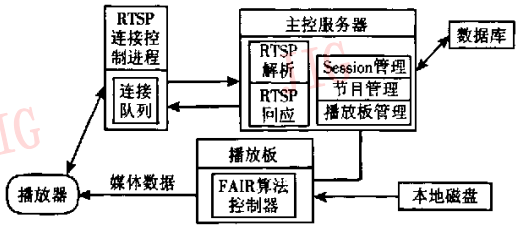

图 1 Clear 点播服务器架构图

Fig. 1 Architecture of Clear VOD server

主控服务器进程用于管理和协调其他进程的运行,相当于任务调度模块,该进程通过访问数据库获得节目发布信息,并维护服务器运行时的信息;RTSP 连接控制进程负责与客户端的 RTSP 交互信息的传输,具有实现相应协议的功能;播放板进程负责服务器的输入/输出。该进程通过访问本地磁盘或磁盘阵列,先得到媒体文件内容,再经由网络设备发送到客户端的播放器。

主控服务器进程通过不同通信方式统筹管理其他进程,这种主从式的架构不仅可以方便地添加功能模块,而且对整体系统影响小。

在优化设计架构时,为对抗恶意攻击,提高服务器健壮性,可将 RTSP 解析模块(对应于图 1,主控服务器下的解析)从主控服务器中剥离,独立为一个进程。它是通过消息队列与主控服务器、RTSP 连接来控制进程通信。由守护进程监控状态,一旦解析进程退出,则立刻重起解析进程。由于解析进程的工作模式类似管道,没有任何状态信息,所以它的结束和重起对整个服务器的工作没有影响。由于解析进程能够屏蔽非法的数据包,起到防火墙的作用,因此可提高服务器的健壮性。

为提高服务器的兼容性,可采用不同进程模块处理不同客户端的方式,将 RTSP 回应模块(对应于图 1,主控服务器下的回应)从服务器中剥离。对于需要支持的新客户端,只需设计开发一种新回应模块,就可在不影响服务器总体架构的前提下,方便地扩展功能。同时,也方便了功能定制,只需根据客户端要求,启动对应的 RTSP 回应模块即可。

最终,笔者设计的服务器采用了这种各个与操作系统无关的模块独立运行并协同工作的主从式架构,在 Linux 和类 Unix 平台下获得了实现,并支持了当下流行的 WMV 格式。该服务器于 2006 年,在

上海市档案馆流媒体点播系统中获得了实际应用。

3.2　提高性能的算法

3.2.1　RTSP 连接队列查询方法

由于 IPTV 应用的用户数庞大,因此 RTSP 连接控制进程需要维护上千的用户队列。若顺序轮询和查找用户,其计算复杂度为 $O(n)$,性能低下。在单 CPU 下会直接影响到服务器其他进程的正常工作。

为了提高查找效率,本系统采用了数组加单链的数据结构。该结构由数组作为申请的可用空间,由一条链表维护数组上的可用空间,记为空闲链表;另一条链表维护已连接的用户,记为用户链表。两条链表都记录了每个表项在数组上的索引。其相比原来的架构,添加、删除和轮询不但没有提高计算复杂度,而且引入数组还省去频繁的内存申请、释放操作。对于查找操作,若使用数组下标来索引内容,则可使计算复杂度下降为 $O(1)$,这也极大地提高了查找性能。

为了提高轮询的效率,笔者考虑到 RTSP 协议的特性——交互发生的频率很低,且主要发生在点播的初始阶段。因此,可设计一种优先级链表用于维护那些更优先的用户。这里定义优先用户是指那些刚刚得到服务器回应的用户,而那些刚刚发出请求尚未得到回应和那些已完成初始交互的用户(一段时间没有请求),则优先级一般。对用户分类以后,链表组织结构如图 2 所示,在轮询时,可以更多关注优先用户,以提高效率。

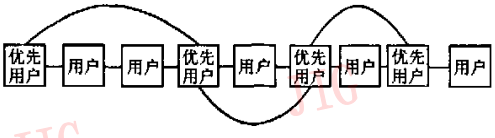

图 2　优先链表结构图

Fig. 2　Structure of priority list

在每 x 次(x 为轮询频度比)轮询优先用户链表中,应轮询一次整个用户链表以保证公平性。这样既提高了服务器重载时的 RTSP 响应时间,也降低了 RTSP 连接控制进程的 CPU 占用率。x 值可根据实际硬件情况和权衡 CPU 和响应时间后设定。

3.2.2　WMV 文件读取和网络发送策略

WMV 是流式组织的格式,其文件内容定位读取方便。播放板采用直接读取 WMV 文件的方式,使用 WMV 的索引数据。WMV 的索引数据结构简洁,其不仅适合驻留内存,而且数据量对于服务器来说也可以接受。

对于架设专用网络的 IPTV 应用来说,瞬时码率过高会引起丢包。为此,本服务器先采用 WTS 算法[3](基于小波的速率平滑算法)使发送速率平滑,且不会引起客户端缓冲的上溢和下溢[4];然后通过对 WMV 文件数据包的发送时间进行微调,即可达到发送速率平滑的效果。同时,服务器通过减少每一次发送的数据量和增加单位时间内的发送次数,以便更微观地平滑发送速率来匹配能力较弱的机顶盒客户端。

由于整个服务器系统的性能瓶颈在于磁盘与网络输入/输出(I/O)的性能,播放板的能力决定了服务器的能力,而主控进程的负载则相对较轻,因此,为了达到性能的平衡与线性扩展,可设计由一个主控进程管理多个播放板进程。由于运行在多台机器上的播放板可通过套接字连接到同一台主控服务器,由主控进程进行负载均衡,因此可以方便地通过添加播放板来线性扩展服务器的性能。

4　系统性能分析

4.1　优先链表分析

设 ps 时间内没有 RTSP 请求的用户,将退出优先用户链表;用户每次点播平均持续时间为 ds;这个 ds 过程中,包括建立会话和用户操作,如果实际 RTSP 交互平均次数为 r 次,则用户处于优先链表中的平均时间是 $r \times ps$。那么对于 N 个用户的并发来说,实际优先链表长度的平均值为 $N \times d/r \times p$。

参考 3.2.1 节,采用优先队列后,相对于原本 x 次轮询长度为 N 的用户链表的操作,实际的操作次数的比值是

$$\frac{(x-1)\dfrac{N \times r \times p}{d} + N}{x \times N} = \frac{r \times p \times x - r \times p + d}{d \times x}$$

由此可以看到,当轮询频度较 x 的取值变大时,实际节省的操作是很显著的,这样也就减少了 CPU 的消耗;考虑到响应时间,实际实现中,轮询频度较 x 不能过大,一般取值在 10 数量级。

4.2　响应时间和吞吐量

RTSP 交互规范,使用微软公司的 7 次交互规范。每次交互在服务器端需经历 4 次消息队列传递,2 次交互需数据库操作,另外还有些必要的计算(轮询用户等)。本文定义响应时间是从用户点击

播放按钮到看到第 1 帧画面所需的时间。具体响应时间如表 1 所示。

表 1 响应时间

Tab.1 Response time

	网络延时	消息队列	DB 操作	其他计算	预存
次数	14	28	2	7	1
耗时(ms)	100	1	10	1	3000

从表 1 可以看到,因为建立会话的整个过程在服务器端耗时小于 20ms,所以响应时间是由网络延时决定。服务器在重载的情况下,能达到每秒建立 50 个会话的吞吐量。考虑到一些已连接用户的交互操作,实际应用所能达到的吞吐量约为每秒建立 30 至 40 个连接,即10000个连接大约只需要 300s 左右的时间,这也就达到 IPTV 应用所需的并发性能。

4.3 稳定性

实际应用中,由于播放板不对外开放端口,因此不会受到攻击,其稳定性主要取决于硬件稳定性。各个播放板可以做到互为备份,以达到较高的稳定性。由于主控服务器需要开放 RTSP 监听端口,其可能受到 RTSP 的恶意攻击,因此本系统采用独立的 RTSP 解析模块,以增加整个主控服务器系统的稳定性。守护进程从监测到解析模块异常退出到重起新的解析模块,其所需要的时间在一个进程切换的时间单位,通常为几 ms;无状态信息的解析模块的初始化时间一般小于 1ms。相对于数十至数百 ms 的 RTT 时延而言,解析模块重起的耗时可以忽略。

5 系统性能测试

在本文设计的系统中,由于主控服务器没有对每个用户起线程,系统不受线程数量的限制。因此,在系统性能测试中,应主要关注 RTSP 队列轮询性能和播放板吞吐性能的测试。

5.1 优先队列性能测试

当测试 500 个用户逐渐连接上服务器,并进行交互时,服务一个用户需要遍历队列的长度,为此本文取轮询频度比 x 值为 10。实验结果如图 3 所示。

由图 3 可见,单队列数值基本在理论值 $N/2$ 附近,优先队列明显优于单队列,从而达到了减少操作,降低 CPU 消耗的作用。

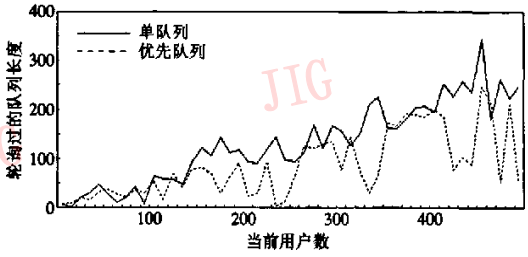

图 3 轮询的比较

Fig.3 Compare of query

5.2 播放板吞吐量性能测试

测试环境,两台服务器双通道 P4 3G CPU,2GB 内存,两个千兆网卡。存储设备为 sata 磁盘阵列,操作系统为 Linux。测试文件码率 2Mbps。

由表 2 可见,由于影片码率有波动,因此更应关注的是网卡的吞吐量,在这种情况下,单块千兆网卡的吞吐量近 90MB,已经接近千兆网卡的极限数字了。

表 2 单网卡单部 2Mbps 影片测试结果

Tab.2 Test result of one file and one network interface

用户	CPU(%)	网卡吞吐量(MB/s)	磁盘吞吐量(KB/s)	效果
301	约 20	75 上下	2048	正常
401	约 30	85 ~ 90	2048	正常
411 +	约 30	接近 90	2048	跳帧

由表 3 可以看到,服务器能够将用户请求均匀地分配在两台播放板上,以达到负载均衡的效果。由于 CPU 的限制,使并发用户数稍低于 800,因此基本可以认为是线性扩展的性能。

表 3 双网卡单部 2Mbps 影片测试结果

Tab.3 Test result of one file and two network interfaces

用户	CPU(%)	网卡吞吐量(MB/s)	磁盘吞吐量(KB/s)	效果
701	约 55	65165	2048	正常
751	约 65	70170	2048	正常
751 +	65 +	共 145 +	2048	跳帧

由表 4 可以看出,磁盘的 I/O 速率大约在每秒 80MB,先于网络吞吐量到达系统性能的瓶颈。由于磁盘每秒 80MB 的读取速率与单块网卡的吞吐量相近,因此单机单网卡的配置下能够更有效地利用系统的硬件资源。

表 4　双网卡多部 2Mbps 影片的测试结果

Tab.4　Test result of multi-files and two network interfaces

用户	CPU(%)	网卡吞吐量(MB/s)	磁盘吞吐量(KB/s)	效果
301	约 25	35135	65	正常
361	约 30	42142	80	正常
361 +	约 30	共 80 +	80	跳帧

对于需要大并发的应用来说,需要一定策略来提高磁盘的 I/O 能力;同时需要合理地运用缓存技术[5]等,以使得整个系统不会出现明显的瓶颈。

6　结　论

由于宽带网络的发展推动了流媒体系统的应用,为此本文针对现有 IPTV 点播系统中一些缺陷和需求之间的矛盾,自主设计,并在 Linux 和类 Unix 平台下实现了支持主流 WMV 文件格式的 Clear 流媒体服务器。该系统提出和采用一系列的架构、调度、数据流读和发送等方面的方法,提升了整个系统的性能。目前系统已成功应用于上海市档案馆档案视频资料点播系统,并作为其中的一部分获 2005 上海市科技进步奖二等奖。在已实现的该系统的基础上,如何解决现有点播系统中的磁盘吞吐量瓶颈的问题将作为后续工作展开研究。

参考文献(References)

1　YE De-jian. Video Quality and Rate Control for Video Streaming Applications [D]. Beijing: Tsinghua University, 2003, (in Chinese). [叶德建. 流媒体系统的视频质量与发送速率控制研究[D]. 北京: 清华大学, 2003.]

2　Etsion Yoav, Tsafrir Dan, Feitelson Dror G. Effects of Clock resolution on the scheduling of interactive and soft real-time processes [A]. In: Proceedings of ACM International Conference on Measurement and Modeling of Computer Systems [C], New York, USA, 2003: 172 ~ 183.

3　Ye De-jian, Barker J C, Xiong Zi-xiang, *et al*. Wavelet-based smoothing of VBR video traffic [J]. IEEE Transactions on Multimedia, 2004, 6(4): 611 ~ 623.

4　Etsion Yoav, Tsafrir Dan, Feitelson Dror G. Process prioritization using output production: Scheduling for multimedia[J]. TOMCCAP Archive, 2006, 2(4): 318 ~ 342.

5　YE De-jian, Zhang Zuo, Wu Qiu-feng. A receiver-buffer-driven approach to adaptive internet video streaming[J]. IEE Electronics Letters, 2002, 38(22): 1405 ~ 1406.