

基于 FTP 协议的素材迁移服务器的设计与实现

詹金华, 刘 锋

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘 要:文中提出的基于 FTP 协议的素材迁移服务器,与传统的采用 FTP 协议服务器和客户端进行素材迁移不同,该迁移服务器对传统的迁移模式进行了改进,让客户端从参加数据传输的角色转变为只对数据传输进行控制的角色。很大程度上减少了客户端的工作强度,符合目前流行的“瘦客户端”的软件设计模式。并且在原来的 FTP 服务器的基础上添加了文件传输代理模块,使客户端不再直接与服务器端进行联系,所有的活动请求都通过代理模块执行。

关键词:FTP 协议;素材;迁移;代理;视频服务器

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2008)03-0056-03

Implementation and Design of Material Transfer Server Based on FTP

ZHAN Jin-hua, LIU Feng

(School of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: The material transfer server based on FTP is different from the traditional model of FTP server and FTP client. In this paper, a great improvement was made on the basis of traditional model so that achieved the goal the role of client changed from execute data transfer to control the data transfer. To some extent, reduced the task of client, it's accord with software model of "thin client". Added the module of the agent so that all requests via the agent module instead of directly contact with the server module.

Key words: FTP; material; transfer; agent; video server

0 引 言

自从 20 世纪末,数字电视在国内得到了很大的发展。随着广播电视行业从原来的模拟电视到数字电视的飞跃发展,原来以录像磁带为记录载体的数据,转变为以计算机为载体的数字音视频文件。广电行业将音视频文件称之为素材。把素材从一台计算机传输给另外一台计算机,称为素材迁移。由于电视行业对画面的质量要求相对较高,因此文件的数据量变得相当庞大。当采集的数据经过线性或非线性编辑后,目前都是通过专用的素材上载工作站上传给视频服务器^[1]。因为对文件传输的可靠性、稳定性和速度的要求相对较高,素材传输大多采用的是 FTP 协议^[2]。笔者在数字电视自动播放控制系统的研发过程中发现,当出现相同的素材迁移到不同的视频服务器时,目前采用的素材迁移方式有一定的缺陷性。目前的解决方法是,

上载工作站对该素材通过多次分别迁移至不同地址的视频服务器,如图 1 所示。无形之中,增加了上载工作人员的劳动量和出错的概率,同时也增大了上载工作站的机器负荷,并且迁移效率低。面对如此情况,提出了利用 FTA(File Transfer Agent,文件迁移服务器代理)解决目前迁移方式中当出现相同素材迁移到多个服务器时的缺陷问题^[3]。

1 FTA 简介

FTA 是在标准 FTP 协议命令的基础上,为了解决所遇到的问题,增加了自定义的控制和传输命令后成功实现的服务器端软件,称之为文件迁移服务器代理,但是它又与服务器模块不完全相同,它包括服务器模块和 Agent 模块。所有的客户端对服务器端的命令请求都必须通过其中的 Agent 来执行,而不能直接与服务器模块进行通讯。服务器模块主要是基于 FTP 协议实现文件的传输功能。FTP 是在 TCP 协议之上实现的一个应用层协议,其主要功能是在计算机之间传送文件。基于 FTP 的软件通常采用 C/S 模式,一个 FTP 服务器进程可同时为多个客户进程提供服务。

收稿日期:2007-06-02

作者简介:詹金华(1979-),男,安徽安庆人,硕士研究生,研究方向为嵌入式多媒体;刘 锋,教授,硕士生导师,研究方向为并行分布计算、计算机网络。

一个FTP连接分为一个控制连接和一个或多个数据连接。控制连接用来在服务器和客户端之间交换命令信息,数据连接实现服务器和客户端的数据交换。在主动模式下,客户程序使用PORT命令告诉FTP服务器自己的数据传输端口号,FTP服务器向这个端口发起一个连接。连接建立后,服务器端和客户端就可以使用这个连接传输数据了。在被动模式下,FTP服务器通过PORT命令告诉客户端使用自己的FTP数据传输端口,然后等待客户端建立数据传输连接^[4-6]。FTP标准命令就不一一列举,现在把文中定义的最为重要的部分命令格式及使用方法列举如下:

* TRANS命令:实现素材文件迁移开始命令。

* PAUSE命令:实现文件迁移的暂停命令。

* STOP命令:取消对文件的迁移命令。

命令格式如下:

TRANS命令

| | | | | | | |
|-----|-------------|----------|---|---------|-----|---------|
| 命令头 | Material ID | SourceIP | N | DestIP1 | ... | DestIPn |
|-----|-------------|----------|---|---------|-----|---------|

PAUSE命令

| | | | | | | |
|-----|-------------|----------|---|---------|-----|---------|
| 命令头 | Material ID | SourceIP | N | DestIP1 | ... | DestIPn |
|-----|-------------|----------|---|---------|-----|---------|

STOP命令

| | | | | | | |
|-----|-------------|----------|---|---------|-----|---------|
| 命令头 | Material ID | SourceIP | N | DestIP1 | ... | DestIPn |
|-----|-------------|----------|---|---------|-----|---------|

说明:Material ID表示素材文件ID,SourceIP表示文件迁移源服务器IP地址,DestIP_n表示素材文件要迁移至的第 n 个服务器的IP地址。 N 表示指定素材文件要迁移到视频服务器的个数。其中,Material ID占2个byte,每个IP地址占用4个byte。当 N 不为零时,可以计算出 N 字段后跟随的命令数据长度 $= N * 4$ byte。当 N 为零时,不用解析 N 字段后的命令数据,迁移的目的视频服务器地址是通过Agent模块在配置文件中找出内部网络内所有的视频服务器的IP地址。

2 迁移服务器的设计与思考

目前上载工作站/服务器工作模式纯粹是点对点文件传输方式在素材文件迁移中的应用。为解决文中前面所指出的缺陷,文中要设计的素材迁移服务器必须实现在同一个时刻向多个视频服务器迁移相同或不同文件,即改进为一对多方式。其次,还应该考虑到工作效率问题,因为数字电视媒体对画面的质量要求较高,因此数据量相当庞大,按照目前每秒钟播放的数据流量为10Mb计算,如果一个频道一天播放15小时,一周的数据量为 $7 * 10 * 60 * 60 * 15 / 8 / 1024 = 461.3G^{[7]}$ 。如此海量数据必须在比较短的时间内传输结束,迁移效率是一个很重要的技术指标。另外在设计过程中还要注意的一个很重要的问题是,必须考

虑迁移服务器对系统资源的占有率问题。因为电视台播控软件在播放节目的时候也不断地从视频服务器中提取数据,在视频服务器的数据吞吐量一定的情况下,若迁移素材过程占用太多的系统资源必然影响节目播出的正常进行。因此首先要解决的问题是,当出现相同素材迁移至不同的服务器时,不是通过素材上载工作站多次地向不同服务器间迁移文件,而应该是在接收到数据迁移命令后,从某一个视频服务器上取出素材数据后在同一时刻以广播方式对多个服务器进行数据发送。基于此,文中对原来的工作模型进行如下改进:其中图1为目前的工作结构图,图2为改进后的结构图。

从结构图1可以看出,当出现相同素材向不同视频服务器之间的迁移时,目前的迁移方式相当不方便,只能从一个固定的上载工作站多次上传文件,而本文提出的迁移方式相对来说更加灵活:只需要从素材上载工作站上载一次给某台视频服务器,然后利用FTA客户端发送迁移命令实现素材在视频服务器之间的迁移。目前的传输方式改进为采用FTA的方式。值得一提的是,FTA客户端不一定要安装在固定的素材上载工作站机器上,而是可以放置在任何一台授权的机器上,这样提高了工作的灵活性。FTA Client部分只用实现控制命令的发送和迁移状态字的接收,因此减少了客户端机器的工作量。当素材迁移源视频服务器断接收到迁移命令后,从磁盘阵列中读取数据发送给目的视频服务器,并且返回当前状态字。

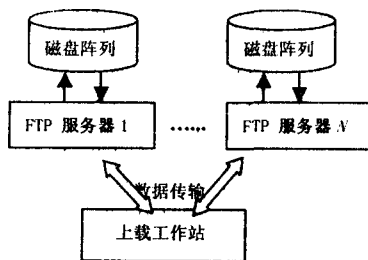


图1 改进前的素材迁移方式结构图

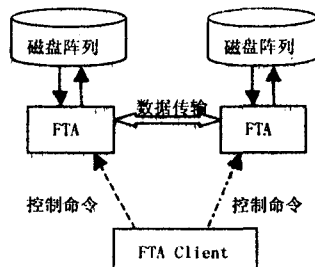


图2 改进后的素材迁移方式结构图

3 系统实现

由前面的分析可以得知,必须考虑资源的占有率

问题。为解决这个问题,采用了多线程的方式,并且在 FTA 中实现了数据传输速度的控制。由一个线程负责监控当前设备资源的使用率,当达到一个阈值时,就降低数据发送速度的阈值和暂停部分负责传输数据的线程。源视频服务器的 Agent 接收到素材迁移命令后,源视频服务器相当于图 1 中的上载工作站,即客户端。因此,在改进后的服务器端也应该包含客户端的功能,因为在素材迁移过程中,每个视频服务器有可能在同一个时刻既作为源服务器也可能作为目的服务器。因此,可以把客户端的功能封装成一个 DLL 或 COM,再与服务器部分融合在一起,再在外面加上一层功能管理模块。而服务器功能部分,文中的解决方案采取模仿 FTP Server 的工作方式,先建立一个线程一直负责监听 FTA Client 的连接。当收到迁移命令后,立即建立数据传输线程实现数据的传输^[8,9]。

对于音视频文件,根据数据流的组织方式,可以分为 ES 流和 PS 流。其中 ES 流表示数据的音频数据和视频数据是分开存放的。由于不同的播控软件可以选择不同格式的多媒体数据流,因此,应该在服务器端包含数据流打包和分离模块,实现 ES 流和 PS 流的相互转换^[10]。

4 性能测试

在相同设备的视频服务器和客户端上,对两种素材迁移方式的性能做了比较。在实验中,以百兆的局域网络,CPU 为 P4 2.8G,512M 内存的计算机为测试平台,以 1.8G 的文件对两种传输方式的速度进行了测量比较,所得相关数据见表 1。

表 1 性能测试

| 目 标 时 间 方 式 | 迁移至一个服务器 | 迁移至 4 个服务器 |
|-------------------|-----------|------------|
| | 改进前方式 | 改进后方式 |
| 改进前方式 | 15 分 23 秒 | 56 分 17 秒 |
| 改进后方式 | 9 分 56 秒 | 18 分 42 秒 |

在资源利用率方面,传统的素材上载工作站与视频服务器一直进行数据的传输工作,视频服务器和上载工作站的 CPU 利用率都比较高。而文中提出的解决方案中,因为客户端只是负责发送控制命令及通过控制命令取回素材迁移的进度和状态信息等,因此 CPU 的利用率明显有很大的下降,而在服务器端,当对 FTA 设置阈值后,系统会一直监控当前资源利用率,确保 CPU 和内存不会超越警戒值。

图 3 为改进前方式的迁移时上载工作站 CPU 曲线图,图 4 为改进后的迁移时利用 FTA 客户端进行迁移控制的工作站 CPU 曲线图。

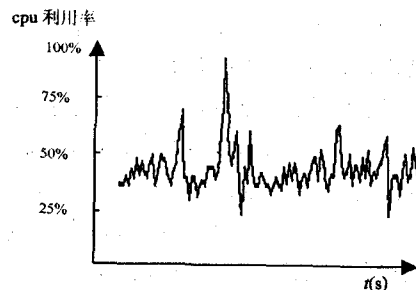


图 3 改进前上载工作站 CPU 曲线图

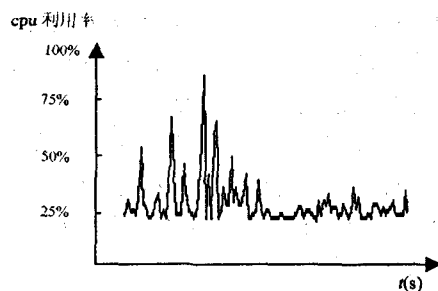


图 4 改进后 FTA 客户端工作站 CPU 曲线图

经过实验观察,在利用素材上载工作站进行迁移时,CPU 在整个过程中,大部分时间保持在 40% 左右,而用文中提出的解决方案时,FTA 客户端的 CPU 利用一直在 20% 以下,其中主要机时是花费在迁移过程中取迁移的进度和状态信息。

图 5 是改进前的视频服务器 CPU 曲线图,图 6 是改进后视频服务器 CPU 曲线图。结合实现算法可以看出,图 6 中的曲线明显比图 5 相对平滑,因为在程序实现部分使用了资源利用率的阈值监测,当快逼近这一值时立即采取措施对数据迁移速度进行调整。而且性能明显优于改进前的。

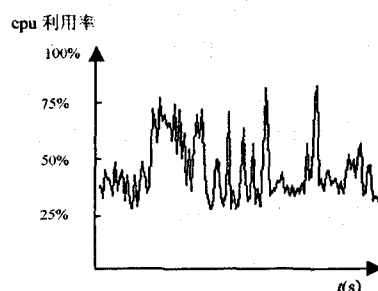


图 5 改进前视频服务器 CPU 曲线图

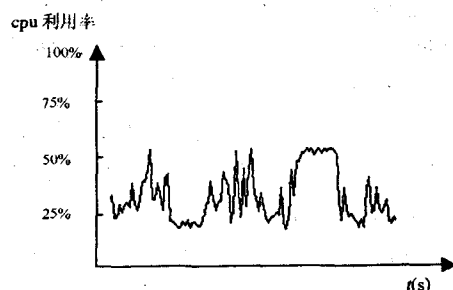


图 6 改进后视频服务器 CPU 曲线图

(下转第 62 页)

式的一个很好近似,但由于在 802.16 WMAN 这样的系统中,一个 BS 域的 SS 的数目非常大,每个 SS 又有成千上万个终端用户,用 Erlang B 公式计算阻塞概率就需要花费大量的时间,而所提出的算法就克服了这一缺点。

图 5 给出了 $\alpha = 1.5, \beta = 0.25$ 时 rtPS 和 nrtPS 服务流平均到达速率和突发阻塞概率之间的关系,其中 SS 的个数为 2000,从图中可以看出,突发阻塞概率将随着平均到达速率的增长而递增,且递增速度逐渐减慢。

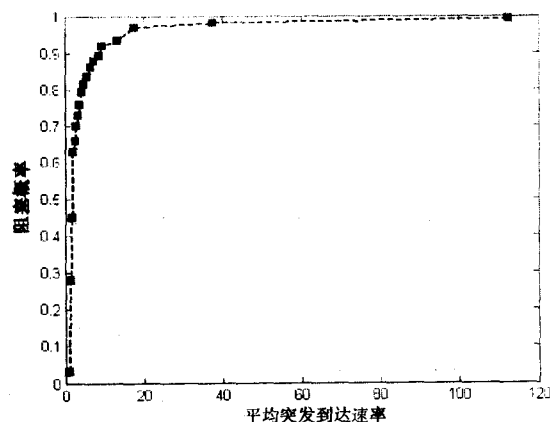


图 5 rtPS 和 nrtPS 服务流阻塞概率曲线

4 结束语

在研究 802.16 系统的 QoS 机制和流模型的基础

(上接第 58 页)

其中图 3 至图 6 中的横轴为软件运行稳定后的时间轴,在运行无任何干涉的情况下,曲线图几乎近似于周期函数曲线。

5 结论

基于 FTP 协议的素材迁移服务器解决了目前上载工作站进行单方向的素材迁移的弊端,在很大程度上提高了工作效率。该素材迁移服务器具有很好的可靠性,并且在资源利用率和数据传输速率方面能实时地做出均衡决策,该服务器现在已经在某些市级电视台投入使用,并且达到了预期设计目标。

参考文献:

- [1] 巫荷才. 校园有线电视硬盘播出系统的设计与构建[J]. 中国有线电视, 2006(9): 947-949.
- [2] 钱建生, 程德强, 田 隽, 等. IP 网络中实时视频流的可靠传输控制[J]. 煤炭科学技术, 2006, 34(5): 53-55.

上,推导出阻塞概率上限和带宽的关系,提出了 802.16 无线 MAN 分组层上的一种带宽分配的机制,并通过仿真将这种阻塞概率上限算法和只能应用于语音流的 Erlang B 公式作了比较。结果证明,阻塞概率上限算法可以用来近似计算大型网络中的阻塞概率,并且在数值计算上比 Erlang B 公式更具高效性。

参考文献:

- [1] IEEE 802.16-2001. IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air interface for fixed broadband wireless access systems[S]. 2002.
- [2] Leland W E, Taqqu M S, Willinger W, et al. On the self-similar nature of Ethernet traffic[J]. IEEE/ACM Transactions on Networking, 1994, 2(1): 1-15.
- [3] Choe J, Shroff N B. A central limit theorem based approach to analyze queue behavior in ATM networks[C]//The 15th International Teletraffic Congress. Washington D. C.: [s. n.], 1997.
- [4] Addie R G, Mannersalo P, Norrors I. Performance formulae for queues with Gaussian input[C]//Proceedings of ITC 16. Elsevier, Edinburgh, UK: [s. n.], 1999: 1169-1178.
- [5] Huo Di, Eurokom M, Dusseldorf. Generalized Erlang-B formula for mobile and wireless radio channels[C]//Global Telecommunications Conference, 1995. Conference record. Communication Theory Mini-Conference, GLOBECOM'95. [s. l.]: IEEE, 1995.

- [3] 郑洪方, 王玉峰, 王光兴, 等. 基于 Mobile Agent 的网络管理系统的流量分析[J]. 小型微型计算机系统, 2006, 27(2): 257-259.
- [4] 余澄丹, 程家兴. 基于 P2P 协议的文件服务器技术[J]. 微机发展, 2005, 15(6): 158-160.
- [5] 陈金阳, 蒋建中, 张良胜. FTP 协议分析及其客户端程序实现[J]. 计算机工程与应用, 2005, 41(1): 130-132.
- [6] 戚伟世, 蔡安妮, 孙景鳌. 基于在线测量的视频服务器接纳控制[J]. 计算机工程与应用, 2005, 41(2): 84-86.
- [7] 苗艳超, 杨晓君, 周应超, 等. 基于 CATV 网络视频服务器的数据实时性保证策略[J]. 小型微型计算机系统, 2006, 27(2): 354-358.
- [8] 樊 华, 朱 莉, 王媛妮. 在基于 H. 323 的视频会议中实现负载均衡[J]. 计算机技术与发展, 2006, 16(4): 122-124.
- [9] 徐朝晖, 钱朴慧. UDP 协议的海量信息快速传输解决方案[J]. 火力与指挥控制, 2005, 30(1): 46-49.
- [10] 任宝宝, 刘彦明. 视频转码技术在网络多媒体传输中的应用[J]. 计算机技术与发展, 2006, 16(6): 86-88.