

适用于 IPTV 大并发应用的文件格式

黄澄, 陈荣华, 叶德建

(复旦大学软件学院, 上海 201203)

摘要: 分析交互式网络电视(IPTV)大并发应用的特性, 提出一种适用于 IPTV 大并发应用的服务器内部文件格式 c14 文件格式。该文件格式采用了符合 IPTV 大并发应用特性的音视频数据交织、媒体数据块状存储等组织策略。设计并实现了一种基于 c14 文件格式的 IPTV 服务器。性能对比实验数据表明, 使用 c14 文件格式能够使 IPTV 服务器拥有更优的并发性能。

关键词: 交互式网络电视; 大并发; 文件格式

File Format for IPTV Large Concurrency Applications

HUANG Cheng, CHEN Rong-hua, YE De-jian

(Software School, Fudan University, Shanghai 201203)

【Abstract】 By studying several key characteristics of Internet Protocol Television(IPTV) large concurrency applications, a media file format named c14 is proposed. In order to work well in IPTV large concurrency applications, this file format uses several media indexing and storage strategies, including video/audio mixture storage and block-level media storage. An IPTV server using c14 file format is designed and implemented. Detailed performance evaluation and test results demonstrate that IPTV server can work more efficiently by using c14 file format.

【Key words】 Internet Protocol Television(IPTV); large concurrency; file format

1 概述

IPTV作为近年来兴起的技术, 是多媒体技术和网络技术的交叉产物^[1]。通过使用IPTV服务, 人们足不出户就能收看成千上万的媒体内容^[2]。随着计算机、机顶盒等终端的普及, IPTV用户数量呈飞速增长趋势。截至2007年上半年, 上海市IPTV用户数目已超过 1.5×10^5 。这样的市场发展趋势要求服务器具有能够适应大并发应用的能力。所谓IPTV大并发应用, 是指IPTV服务器必须能够同时为成千上万的用户提供实时、流畅的服务。

面对IPTV大并发应用, 运营商可以采用服务器集群的方式提高IPTV系统整体用户并发上限, 然而简单地增加服务器数量会大大增加运营成本。因此, 充分发挥服务器集群中每台服务器的软硬件能力、提高其并发上限是提高IPTV系统整体用户并发上限的首选方案。

目前业界存在种类繁多的文件格式, 很多文件格式并没有为IPTV大并发应用进行特别设计。为了实现高性能IPTV服务, 必须针对IPTV大并发应用特性对媒体文件索引、搜索定位、存储方式进行特别的设计。

2 问题描述

IPTV大并发应用有着下面3个特性:

(1)流媒体特性

IPTV大并发应用是一种流媒体应用, 也具有流媒体“边下载边观看”的特性。用户点播一部影片之后, 只需要等待短暂的缓冲时间就可以进行观看, 而不必等到整部影片都下载完毕。

(2)高流量数据 I/O 特性

IPTV大并发应用中, 服务器必须从影片库中提取大量的媒体数据发送给大量用户^[3]。因此, 该应用有着高流量数据I/O的特性, 其中包含磁盘I/O和网络I/O。

(3)媒体数据实时性和连续性

IPTV服务器必须向用户发送实时的、具有时间连续性的媒体数据, 即这些数据是用户当前所需的、在影片中对应的的时间戳是连续的^[4]。

为了更好地适应上述特性、提高服务器并发上限等性能, IPTV大并发应用中使用的文件格式必须具有下面的特点:

(1)包含IPTV大并发相关的、便于提取的索引信息, 如发送时间戳信息、媒体类型信息、数据位置/大小信息等。

(2)便于提取时间信息连续的数据。

(3)便于进行高流量数据 I/O, 主要涉及磁盘 I/O。

3 c14 文件格式

本文提出了一种针对IPTV大并发应用特性设计的服务器内部文件格式c14文件格式, 如图1所示。



图1 c14 block 结构

c14文件由若干个数据集块block组成, 每一个block都具有相同的总大小和相同的组织结构。如图1所示, 每一个block分为索引信息区和媒体数据区两部分。其中以音视频帧为单位分别存储媒体数据的索引信息和实际数据, 而且2块区域中的“帧索引”和“帧媒体数据”是一一对应的。索引

基金项目: 国家自然科学基金资助项目(60503044)

作者简介: 黄澄(1983—), 男, 硕士研究生, 研究方向: 流媒体与宽带网络; 陈荣华, 讲师; 叶德建, 副教授

收稿日期: 2007-10-25 **E-mail:** 042053023@fudan.edu.cn

信息包含了 IPTV 数据发送相关的发送时间戳信息、媒体类型信息、数据位置/大小等索引信息。

为了提高 IPTV 服务器的大并发性能，c14 文件的设计中采用了音视频数据交织存储、媒体数据块状存储策略。

3.1 音视频数据交织存储

服务器需要根据客户端要求同步发送视频数据和音频数据，即按照发送时间顺序依次将视频数据和音频数据发送给客户端。这样用户才能观赏到图像和声音同步的影片。基于该特点，c14 文件中采用了音视频数据交织存储策略：以音视频帧为最小单位，按照发送时间顺序对音视频数据进行混合存储。向客户端连续发送音视频数据时，服务器只需从 c14 文件中顺序读取 block，并按照 block 内部记录的发送时间进行发送即可。这样，服务器进行连续数据发送时避免了音视频同步操作延迟。此外，c14 文件将原本分开存储的音视频数据交织混合存储，使得服务器能够顺序读取磁盘文件，提高磁盘的读取性能，进而提高服务器提取媒体数据效率。

3.2 媒体数据块状存储

c14 文件中采用了音视频数据交织存储策略，因此服务器进行连续数据发送时只需要对文件进行顺序读取。为了提高服务器进行连续数据发送时的磁盘 I/O 效率，笔者进一步在 c14 文件格式设计中采用了媒体数据块状存储策略，同时在服务器中采用了媒体数据块状读取策略。媒体数据块状存储策略是指：将时间连续的媒体数据（以音视频帧为单位）和相应的索引信息集中组织，形成块状数据集群 block 在媒体文件中进行存储。媒体数据块状读取策略是指：服务器以 block 为单位将磁盘数据读入缓冲区，然后根据 block 中索引信息提取当前发送计划需要的音视频数据。由于 block 当中包含了一段时间内的音视频数据及其相关的 IPTV 索引信息，因此服务器进行一次 block 读取操作就可以满足向某用户连续发送一段时间媒体数据的需求。这样可以减少磁盘 I/O 请求总数，达到提高磁盘 I/O 效率的目的。根据实验比较，采用 1 MB 作为 block 的总大小。

4 基于 c14 的 ClearServer

ClearServer 服务器采用 c14 文件构建影片库，避免了不同文件格式对应音视频数据提取过程的巨大差异性，简化了服务器媒体数据发送模块的工作流程。图 2 显示了服务器各主要模块以及它们之间的数据流向关系。

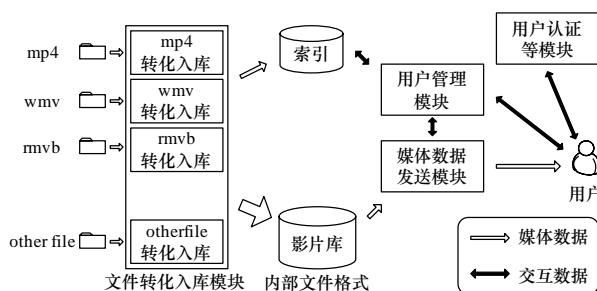


图 2 VoD 系统整体架构

用户管理模块负责接收客户端的交互数据，对所有用户的状态进行管理，并从宏观层面指导媒体数据发送模块进行媒体数据发送。

媒体数据发送模块按照用户管理模块提供的用户状态信息制定整体数据发送计划，并按照发送计划从影片库中提取音视频数据发送给其分管的所有客户端。

文件转化入库模块负责将送入服务器影片库的不同格式

媒体文件转化成为 c14 文件，并生成索引信息供用户管理模块和媒体数据发送模块查询。

用户认证等模块进行用户认证等工作，也是 VoD 系统中必不可少的部分。

与直接使用一般文件格式的服务器相比，ClearServer 中增加了一个文件转化入库模块。文件转化入库模块将其他文件转化为 c14 文件，转化过程去除了原先文件中不用于 IPTV 应用的数据。因此 c14 文件只包含与 IPTV 应用相关的音视频数据以及必要的索引信息，能够减少服务器的数据筛选工作。文件转化流程分为两部分：从源文件中提取所有需要发送给客户端的音视频数据及其对应的发送时间戳等必要索引信息；将提取出的数据封装生成 c14 文件。数据提取过程实质上就是其他直接使用源文件的服务器在数据发送时需要反复进行的数据提取过程。ClearServer 将此过程分离出来预先处理，在数据发送时直接从更加高效的 c14 文件中提取数据。虽然在文件入库时增加了一些文件转化时间，但能够有效提高服务器在数据发送时的数据提取效率，有效提高服务器的并发上限。

由于统一采用 c14 文件格式进行影片库构建，ClearServer 便捷地实现对多种文件格式的支持。服务器目前支持 mov/mp4, wmv, MPEG-2TS/PS(Transport Stream/Program Stream)等文件，而且可以通过扩展文件转化入库模块便捷地增加对 rm/rmvb 等其他文件的支持。

5 性能对比分析

5.1 块状数据 I/O 性能对比

实际磁盘读写过程包括读写请求传递过程、磁头搜索过程和数据处理过程。选取不同大小的单次读写数据量 S_{io} ，数据处理过程和其他过程之间的时间比例搭配不一样，服务器的磁盘 I/O 效率会有很大差别。笔者进行了随机写文件实验，分析不同 S_{io} 对应的磁盘吞吐量 TP 、单次 I/O 平均耗时 ST_{avg} 和单次 I/O 最大耗时 ST_{max} 。

由于读写请求传递过程和磁头搜索过程的耗时与 S_{io} 大小无关，因此理论上单次从磁盘读取或写入越多的数据，单位数据量平摊的读写请求传递和磁头搜索时间越小，磁盘的吞吐效率越高。实际上，由于物理磁盘寻道速度和转速的限制，磁盘吞吐效率不可能随着 S_{io} 的增大而无限增大。表 1 中数据表明：当 S_{io} 达到 1 MB 以上时，实验所用磁盘的吞吐量已经基本稳定了。表 1 数据还表明： S_{io} 过大会造成单次读写独占过长的 CPU 时间。单次读写独占过长的 CPU 时间会阻碍服务器其他逻辑的正常运行。

表 1 不同单次读写数据量对应的磁盘性能

| S_{io}/KB | $TP/(MB \cdot s^{-1})$ | ST_{avg}/ms | ST_{max}/ms |
|-------------|------------------------|---------------|---------------|
| 1 | 3.649 | 1.338 | 203.276 |
| 4 | 15.060 | 1.296 | 101.766 |
| 16 | 19.586 | 3.989 | 92.154 |
| 64 | 19.951 | 15.663 | 222.551 |
| 256 | 23.896 | 52.310 | 200.202 |
| 1 024 | 27.966 | 178.788 | 485.358 |
| 4 096 | 27.289 | 732.890 | 904.601 |

本组实验表明：采用 1 MB 作为 c14 文件格式的 block 大小是合理的。

5.2 媒体数据提取流程对比(c14 vs MP4)

由于 c14 文件中采用了适合 IPTV 大并发应用的数据组织策略，ClearServer(简称 C/S 服务器)在进行数据提取时能够获得比直接使用一般文件提取媒体数据的其他服务器更高的提取速度。下面以直接使用 mp4 文件的 Darwin Stream Server(简称为 DSS 服务器)^[5]作为对象进行比较说明。

C/S 服务器单次提取媒体数据进行发送的流程: (1)确定当前发送所需数据对应的 c14 文件 block 序号; (2)读取目标 block 进入服务器内部缓冲区; (3)利用 block 中索引信息提取媒体数据; (4)按照发送协议分包发送。

DSS 服务器单次提取媒体数据进行发送的流程: (1)通过 mp4 文件内 hint track 定位 hint sample; (2)读取 hint sample; (3)分析 hint sample 内容, 定位当前发送计划所需实际媒体数据; (4)读取实际媒体数据; (5)按照发送协议分包发送。

进行单次数据发送时, C/S 服务器需要 4 步, 其中只有步骤(2)是磁盘操作, 其他都是内存操作; DSS 服务器需要 5 步, 其中包含(2)和(4) 2 次零散磁盘操作。

IPTV 大并发应用中的媒体流具有时间连续性: 在连续发送媒体数据时, 服务器总是按照媒体数据的时间顺序依次提取并发送。在连续提取媒体数据时, 使用 mp4 文件格式需要反复进行上述的过程, 产生的磁盘访问是随机方式的、零散的; 由于采用音视频数据交织存储和媒体数据块状存储策略, 读取一次 block 既可满足多次数据发送, 使用 c14 文件格式产生的磁盘访问是顺序方式的、块状集中的。在目前大多数的磁盘存储介质中, 块状集中化顺序访问的效率远高于零散随机访问。因此, 使用 c14 文件能够更加高效地提取连续音视频数据。这使得服务器在单位时间内能够为更多用户提供实时服务, 也使得服务器的并发性能更优。

5.3 服务器性能比较

笔者对 C/S 服务器和 DSS 服务器进行了并发性能对比。

实验使用下面软硬件配置的服务器: 双至强(Xeon) Pentium IV 3.0GHz CPU(2 颗双核 CPU), 4 GB 双通道内存, 千兆网卡, ST1500 磁盘阵列, RedHat Linux 操作系统(内核版本 2.4)。

实验中所有用户点播影片的编码码率为 2 Mb/s。实验数据经过 Linux 下的软件 dstat 和实验工具的实时数据双重验证。由于服务器程序都位于一颗 CPU 的一个“核”上, 表 2 中“CPU 占用量”是指一颗 CPU 上面的“单核”工作情况。表 2 数据表明, 在当前服务器软硬件环境下, C/S 服务器的并发上限是 410 个流, 而 DSS 服务器的并发上限只有

(上接第 66 页)

别采用 Hu 算法、PSO 算法、QPSO 算法进行实验, 结果如表 1 所示, 其中, 带有*标记的表示最小属性简约。

表 1 属性简约结果

| 数据集名称 | 属性数 | 记录数 | 最小约简 | | Hu 算法 | | PSO 约简算法 | | QPSO 约简算法 | |
|---------------|-----|-----|------|-----|-------|-------|----------|-------|-----------|-------|
| | | | 属性数 | 属性数 | 属性数 | 时间/ms | 属性数 | 时间/ms | 属性数 | 时间/ms |
| Sponge | 45 | 76 | 8 | 8* | 1 016 | 12 | 4 813 | 8* | 6 219 | |
| house | 17 | 90 | 4 | 5 | 565 | 5 | 4 312 | 4* | 4 141 | |
| Zoo | 17 | 101 | 5 | 6 | 610 | 6 | 1 516 | 5* | 1 672 | |
| Lymphography | 19 | 148 | 6 | 6* | 1 621 | 7 | 13 516 | 6* | 13 813 | |
| Soybean_large | 36 | 307 | 9 | 10 | 5 281 | 12 | 133 937 | 9* | 133 703 | |

由表 1 可以看出, QPSO 约简算法比 Hu 算法和 PSO 约简算法取得了更好的约简结果。对数据集 house, Zoo, Soybean_large, Hu 算法都不能得到最小约简结果; PSO 约简算法容易陷入局部最优, 它虽然找到了一组属性约简, 但不是最小约简; QPSO 约简算法找到了最小约简结果。QPSO 约简算法与 PSO 约简算法的耗时相差不多, 略大于 Hu 算法, 这是因为种群要经过多次迭代, 才能得出结果。

261 个流。由于使用 c14 文件, C/S 服务器能够非常高效地获取发送所需的音视频数据, 在数据提取操作上占用较少的 CPU 等系统资源。在同样的客户端压力下, C/S 服务器的 CPU 占用量都小于等于 DSS 服务器的 CPU 占用量。而且随着并发压力的增大, C/S 服务器在节省 CPU 方面的优势愈加明显。DSS 服务器达到并发上限 261 个用户时占用大约 40% 的 CPU, 同样并发压力下 C/S 服务器只占用 20% 不到的 CPU。由此可见, 使用 c14 文件格式对减少服务器 CPU 耗费、提高服务器并发上限具有积极贡献。

表 2 服务器并发数据

| 服务器 | 用户数 | CPU 占用量(%) | 效果 |
|-----|-----|------------|------|
| C/S | 201 | 约 10 | 正常 |
| C/S | 301 | 约 20 | 正常 |
| C/S | 410 | 约 30 | 正常 |
| C/S | 411 | 约 30 | 出现跳帧 |
| DSS | 201 | 约 10 | 正常 |
| DSS | 261 | 约 40 | 正常 |
| DSS | 262 | 约 40 | 出现跳帧 |

6 结束语

本文分析了 IPTV 大并发应用的特性, 有针对性地提出一种适用于 IPTV 大并发应用的 c14 文件格式。实验数据表明, 该文件格式能够有效提高服务器并发上限等性能。继续挖掘 IPTV 应用的特点、进一步完善 c14 文件结构、提高服务器性能将是今后研究方向。

参考文献

- [1] Wales C, Kim S, Leuenerger D, et al. IPTV—The Revolution Is Here[EB/OL]. (2005-07-12). http://www.eecs.berkeley.edu/~binetude/course/eng298a_2/IPTV.pdf.
- [2] 钟玉琢. 多媒体技术[M]. 北京: 清华大学出版社, 1999.
- [3] 王荣生. VoD 系统性能的综合研究[J]. 计算机工程, 2005, 31(4): 11-13.
- [4] Wu Zhengda. Design and Analysis of Video-on-demand Servers [C]//Proc. of IEEE Global Telecommunications Conference. Sydney, Australia: [s. n.], 1998: 773-778.
- [5] Apple Inc. Streaming Server[EB/OL]. (2007-05-20). <http://developer.apple.com/opensource/server/streaming/>.

6 结束语

量子粒子群与属性约简算法相结合可以更容易地找到最小属性约简。但量子粒子群算法属于启发式搜索算法, 不能保证得到的是最优解, 因此, 需要进一步研究改进方法。

参考文献

- [1] Hu Xiaohua, Cercone N. Learning in Relational Database: A Rough Set Approach[J]. Computational Intelligence, 1995, 11(2): 323-337.
- [2] Wang Xiangyang, Yang Jie, Teng Xiaolong, et al. Feature Selection Based on Rough Sets and Particle Swarm Optimization[J]. Pattern Recognition Letters, 2007, 28(4): 459-471.
- [3] Sun Jun, Feng Bin, Xu Wenbo. Particle Swarm Optimization with Particles Having Quantum Behavior[C]//Proc. of IEEE Congress on Evolutionary Computation. [S. l.]: IEEE Press, 2004.
- [4] Sun Jun, Xu Wenbo, Feng Bin. A Global Search Strategy of Quantum-behaved Particle Swarm Optimization[C]//Proceedings of IEEE Conference on Cybernetics and Intelligent Systems. [S. l.]: IEEE Press, 2004.
- [5] Clerc M, Kennedy J. The Particle Swarm: Explosion, Stability, and Convergence in a Multidimensional Complex Space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.