第 34 卷 第 20 期 计 算 机 工 程 2008 年 10 月
Vol.34 No.20 Computer Engineering October 2008
·软件技术与数据库· 文章编号：1000—3428(2008)20—0071—03 文献标识码：A 中图分类号：TP393

P2P 点播系统的客户端磁盘缓存策略

孙名松，唐 亮，周红敏
(哈尔滨理工大学网络信息中心，哈尔滨 150080)

摘 要：在基于有服务器结构对等网络环境的点播系统中，针对客户端需要缓存部分流媒体数据的问题，提出一种新型预测双缓存模型，给出相应的缓存替换算法。仿真实验表明，该模型在用户数目增多时，客户端缓存的命中率大幅提高，服务器的负载得到改善，显著降低了 VCR 操作时所引起的等待延迟。

关键词：对等网络；流媒体；预测缓存；点播

Client Disk Caching Strategy on P2P VoD System

SUN Ming-song, TANG Liang, ZHOU Hong-min
(Network Information Center, Harbin University of Science and Technology, Harbin 150080)

【Abstract】 In the Video on Demand(VoD) system that under the Peer to Peer(P2P) network environment with structure of server and as for the client side's demand on the cache media streaming data, this paper proposes a new prediction double buffer model and gives the corresponding cache replacement algorithm. The model is validated by the simulation experiment that along with the growing number of users, it can greatly raise the hit ratio of client cache, the load of server is improved, and the model significantly lowers the waiting delay during the operating process of Video Cassette Recorder(VCR).

【Key words】 Peer to Peer(P2P); streaming media; prediction caching; Video on Demand(VoD)

1 概述

缓存技术是计算机中普遍采用的基本技术之一，在对等网络(Peer to Peer, P2P)中，缓存技术的主要思想表现为：当发生资源请求时，首先到本地缓存中进行查询，如果命中，返回结果直接响应；否则，用户的请求将仍然通过路由机制导航到存储所需求对象的目的节点上。从 P2P 流媒体直播、点播系统的结构和原理中不难发现，P2P 网络由于受网络结构的不确定性、节点频繁加入、退出网络等因素的影响，导致用户行为表现为不确定性。相对于直播系统来说，点播系统所受的影响将更大，因为只有当同时观看同一部影片同一时间段的人数达到一定规模时，点播才能顺利进行，相反点播人数较少时，不仅不能表现出 P2P 网络的优点，反而可能导致点播视频断断续续而无法观看，这时的性能可能还不如 Client/Server(C/S)架构点播系统的性能。

为解决上述问题，在目前的P2P流媒体点播系统中广泛采用了代理缓存[1]技术，以减少用户访问延迟、降低网络开销、平衡负载及提高服务质量[2-3]。如果将流媒体数据缓存到客户端，能进一步减少路由次数并降低网络开销。因此，客户端中的缓存策略对P2P点播系统的整体性能也会产生一定影响[4]。

2 缓存模型

目前，已经产生并应用于商业模式的 P2P 流媒体点播系统有 2 种：

(1)没有专门服务器的纯 P2P 点播系统；

(2)有目录服务器的混合式 P2P 点播系统。

无论是哪种系统，为了有效利用各客户节点的存储空间和带宽，一般将流媒体分割成若干块，系统以块为单位对点播内容进行查找、下载、播放等操作。本文提出的适合混合式 P2P 点播系统的客户端缓存模型如图 1 所示。

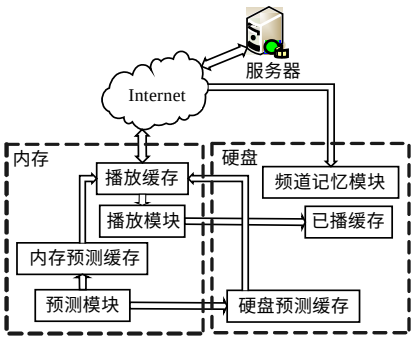

图 1 P2P 点播系统客户端缓存模型

传统的点播实现方案在用户选择某一具体节目进行点播时，需要长时间等待才能观看。点播开始后，用户通常只浏览该视频的起始或终止部分以判断对此视频是否感兴趣，最终决定是否继续观看。因此，在设计缓存模型时加入了频道记忆模块。该模块为服务器提供的频道保存一张与频道对应的 Hash 表。

在点播系统启动初始化时，从服务器或该节点所在区域的超级节点处获得频道列表。当用户点播某一具体频道时，按缓存的列表中提供的信息(信息格式如图 2 所示)直接连接而无须等到路由模块返回，这样可以显著缩短启动延迟。

作者简介：孙名松(1963－)，男，教授，主研方向：网络应用，网络安全；唐 亮、周红敏，硕士研究生

收稿日期：2007-11-13 **E-mail：**tangliangbing@163.com

Channel-ID	StreamingMedia-ID	TheBestList

图 2　频道记忆模块结构

其中，Channel-ID 表示频道号；StreamingMedia-ID 表示流媒体块号；TheBestList 表示目前正在点播该频道、块号的最优节点列表。

预测模块的结构如图 3 所示，分为 3 个部分：(1)短期预测(ShortPrediction)。短期预测的结果直接存入内存中的预测缓存，当进行 VCR 操作产生定位延迟时，先查询内存中的预测缓存，若命中就直接响应；否则查询长期预测部分。(2)长期预测(LongPrediction)。播放媒体的同时，根据 Gossip 协议的时间相关性，在当前带宽负载不严重时，下载部分离正在点播块较远的内容到硬盘进行缓存。在短期预测失败的情况下，可以查询硬盘中的预测缓存。若命中，将其调入内存，通过相应的替换策略进行替换、然后播放；若没有命中，则转到预测仲裁。(3)预测仲裁(PredictionAbiter)。根据当前的网络状态以及对等节点的情况，找出对本地节点最有利的对等节点，在预测失败时，供路由选择模块使用。

ShortPrediction	LongPrediction	PredictionAbiter

图 3　预测模块结构

本文提出的客户端缓存模型中并不将播放完的视频块直接丢弃，而是将其按一定的算法存入硬盘中的已播缓存中。这样在用户进行 VCR、向前拖动时，响应时间极短。

缓存模型的工作流程描述如下：

(1)获取频道记忆列表，并对内存、硬盘缓存空间进行格式化。

(2)用户点播某个节目后，从频道记忆列表中获得最优对等节点进行连接，同时发起路由选择，更新频道记忆列表。

(3)缓存 P2P 流媒体块数据并播放。将播放后的数据缓存到硬盘的已播缓存中。

(4)根据 Gossip 协议的时间相关性原理，启动预测模块。

(5)如果产生 VCR 操作，先查询内存中的预测缓存，命中则转(3)，否则转(6)。

(6)查询硬盘中的预测缓存以及已播缓存，命中则转(3)，否则转(7)。

(7)发起路由选择，用户是否更换其他频道，是转(2)，否则转(3)。

3　替换算法

对于 P2P 流媒体点播的客户端缓存而言，媒体的流行度并不重要。主要应考虑哪些内容需要缓存、哪些内容需要从缓存中淘汰。目前对客户端的缓存的研究并不多，一个较好的客户端缓存算法要替换出使用价值小的、缓存使用价值大的数据，同时还要关注其他节点的缓存情况。针对本文的缓存模型，给出了预测缓存替换算法。播放缓存采用 LRU 算法。

设缓存替代请求时刻为t，记此时预测缓存中的流媒体文件集合为$Cfile=\{1, 2,\cdots,n\}$，文件块i在预测缓存部分的大小为$size(i)$，$i\in Cfile$，缓存容量上限为$Ulimit$，那么应该满足$\sum_{i\in Cfile} size(i) \le Ulimit$。用户点播某个频道，该频道的视频文件被切割成若干份，每份在i时刻被点播的概率为P_i，$i\in Cfile$，则在硬盘的预测缓存中取$\min \sum_{i\in Cfile} P_i$，而在内存的预测缓存中取$\max \sum_{i\in Cfile} P_i$。在点播的过程中，访问某个时间段$\Delta t$内的人数有多有少，即点播的概率$P_i$并不一定相同，点播概率大的时候能体现P2P点播的优点，文件副本比较多，在路由搜索时可以得到很快的响应，因此，内存中的预测缓存采用缓存点播概率较大的文件块。这样可以实时响应VCR操作，缩短响应时间。硬盘的预测缓存则采用相反的替换算法，考虑到Δt时间内点播概率大的文件块可以在内存预测缓存中保存，同时如果产生VCR操作，由于资源数比较多，可以直接进行路由搜索。而当Δt时间内点播概率较小的时候，就不宜等到产生VCR时进行路由，因为此时的资源数少，响应时间肯定会变长。因此，硬盘的预测缓存主要是缓存点播概率小的文件块。在VCR操作发生时，可以先从硬盘的预测缓存中进行搜索，有效解决了因点播资源少而导致长时间下载缓冲、不能观看的问题。替换时将$\max\{P_i \mid i\in Cfile\}$的文件块替换出去。

该算法的伪代码描述如下：

```
ReplaceAlgorithm(待缓存文件块 i){
 If(Pi<阈值){//缓存到硬盘的预测缓存中
        If(size(i)<size(硬盘预测缓存的空闲空间)){
 直接缓存该文件块;}
Else{
 Tempi=find(Pi);//该函数找出当前预测缓存中比Pi大的文件   //
块，将最大值返回。
      替换P_tempi位置的文件块，将文件块i放入缓存;}}
Else{
        If(size(i)<size(内存预测缓存的空闲空间)){
直接缓存该文件块;}
Else{
Tempj=find(Pi);//该函数找出当前预测缓存中比Pi小的文件块，//
将最小值返回。
替换P_tempj位置的文件块，将文件块i放入缓存;}}}
```

4　仿真实验

预测双缓存策略增加的是客户端的磁盘空间，除了频道记忆列表外，没有增加网络的控制信息量，对每次程序初始化时，只增加 1 次与服务器的请求和应答，而且所需传递的信息数量很少，对系统的整体性能基本上不会产生影响。

衡量 P2P 流媒体点播系统性能的主要指标有：(1)系统能同时支持的客户端数量；(2)点播时视频是否流畅。混合式 P2P 流媒体点播系统主要就是利用客户端来分担服务器的负载，在边下载播放的同时给对等节点上传。本文利用模拟实验，通过对比在相同情况下对等节点分别采用预测双缓冲策略和传统不加预测的缓冲策略时，缓存命中次数与缓存大小以及客户端数目之间的关系，来研究预测双缓冲策略的有效性。

用一个离散事件仿真器来模拟系统中的服务器和对等节点在系统运行中的行为，仿真器由用户的 VCR 操作驱动。当产生一个 VCR 时，仿真器随机选择一个对等节点作为请求的发出者，同时服务器将当前点播该频道的该时间段的流媒体块的对等节点列表发送给请求者，请求者接收到后启动缓存算法。

本实验在 Windows 2000, PIII 600 MHz, 512 MB 内存的机器上完成。实验假定有 1 台流媒体服务器，1 个 60 min 长流媒体被分割为 3 600 块，阈值为 0.5。测试用的客户端数目分别为 1, 10, 100, 1 000, 10 000，其中，每台都有缓存能力，即可以按预测双缓存策略提供下载、播放、缓存、上传等服

务。实验中忽略各种事件的处理时间，不考虑实际网络的影响。模拟实验从系统进入稳定状态时(各对等节点均已随机缓存了单个频道的某些媒体块)开始计时。图 4~图 6 给出了部分实验结果。

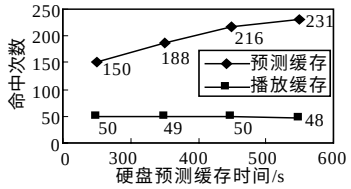

图 4　硬盘预测缓存时间与命中次数关系

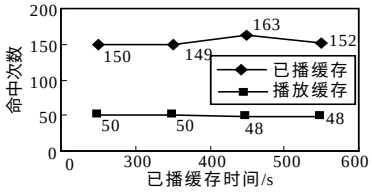

图 5　已播缓存时间与命中次数关系

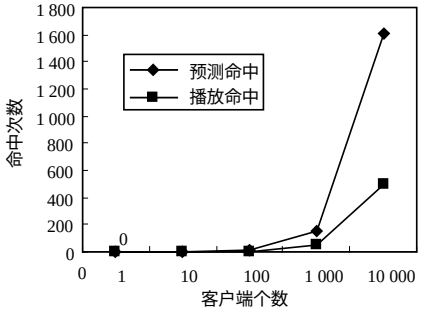

图 6　客户端数目与命中次数关系

图 4~图 5 是在 1 000 个客户端节点、不同的缓存时间的情况下，系统进入稳定状态后，命中次数随着缓存时间的变化曲线。从图 4 中可以看出，硬盘预测缓存时间对命中次数有一定的影响，基本上呈线性增长。从图 5 中可以看出，已播缓存时间对命中率的影响不是很大，因为在观看一部影片，点播刚看完的片段要比点播没有观看的片段的概率小。

图 6 显示了客户端数目与命中次数的关系，在客户端数目较少时双缓冲策略与传统的缓存策略相差不大。当客户端数目比较多时，即点播该影片的人数多时，双缓冲策略明显优于传统的缓存策略，命中次数显著增加。验证了点播的人数越多越能体现 P2P 流媒体特性这一特点。

5　结束语

本文在对传统缓存策略分析的基础上，结合 P2P 流媒体点播系统的特点，设计了一种基于硬盘、内存的预测双缓冲策略。该策略中包含部分媒体流在硬盘缓存中保存，涉及到读写硬盘。以后的工作将进一步探讨是否可以减少硬盘缓存的读写次数，以减小对硬盘的伤害。

参考文献

[1] Liu Yunqiang, Yu Songyu. Streaming Media Delivery with Proxy Cache for Heterogeneous Clients[C]//Proceedings of the 7th IEEE Workshop on Multimedia Signal Processing. Shanghai, China: [s. n.], 2005.

[2] Chang Shin-hung, Chang Ray-I. A Priority Selected Cache Algorithm for Video Relay in Streaming Applications[J]. IEEE Transactions on Broadcasting, 2007, 53(1): 79-91.

[3] Chen Songqing, Shen Bo, Susie W, et al. Adaptive and Lazy Segmentation Based Proxy Caching for Streaming Media Delivery[C]//Proc. of NOSSDAV'03. Monterey, CA, USA: [s. n.], 2003.

[4] Sasabe M, Wakamiya N. Scalable and Continuous Media Steaming on Peer-to-Peer Networks[C]//Proceedings of the 3rd International Conference on Peer-to-Peer Computing. Linköping, Sweden: [s. n.], 2003.

(上接第 70 页)

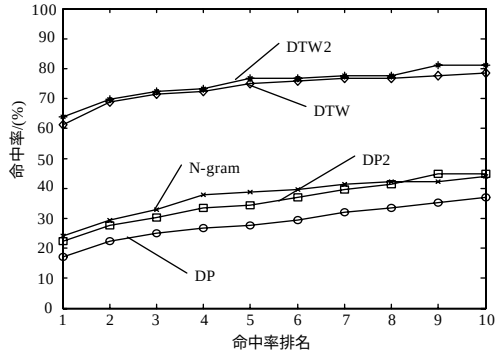

图 6　5 种算法的前 10 位命中率

5　结束语

音高和音长是音符的重要信息，目前大多检索算法将音高和音长信息分开利用，分别计算音高轮廓相似度和音长相似度，然后根据各自的权重综合两者。本文分析了 DTW 算法的不足之处，并将音长信息引入其中，在此基础上实现了一个哼唱检索系统的原型，实验证明了本算法的有效性。在数据库容量扩大的情况下证明本算法的有效性以及提高检索速度是值得人们进一步研究的问题。

参考文献

[1] Ghias A, Logan J, Chamberlain D, et al. Query By Humming Musical Information Retrieval in An Audio Database[C]// Proceedings of the 3rd ACM International Conference on Multimedia. San Francisco, California, USA: ACM Press, 1995.

[2] Wu Yadong, Li Yang, Liu Baolong. A New Method for Approximate Melody Matching[C]//Proc. of IEEE International Conference on Machine Learning and Cybernetics. Xi'an, China: IEEE Press, 2003.

[3] 金　毅, 黄　敏. 基于旋律的音乐检索[J]. 情报学报, 2003, 22(3): 297-301.

[4] Lu Lie, You Hong, Zhang Hongjiang. A New Approach to Query By Humming in Music Retrieval[C]//Proc. of IEEE International Conf. on Multimedia and Expo. Tokyo, Japan: IEEE Press, 2001.

[5] 马志欣, 付少锋, 周利华. 哼唱检索中一种新的旋律模糊匹配方法[J]. 西安电子科技大学, 2006, 33(1): 85-88.

[6] 富　亮. 基于内容的音乐检索研究[D]. 上海: 复旦大学, 2005.

[7] 李　明. 基于哼唱的音乐检索研究[D]. 北京: 中国科学院声学所, 2005.