

# 一种 P2P 流媒体系统中的缓存替换算法

张继荣, 刘艳君

(西安邮电学院 通信与信息工程学院, 陕西 西安 710061)

**摘要:** 针对视频节目受欢迎程度不同的特性, 提出一种 P2P 流媒体系统中的缓存替换算法, 通过将系统中的全部视频片段分类, 为其赋予不同的优先级, 并周期性地更新该值, 同时考虑视频片段被访问次数和最近被访问的情况, 使得被替换出存储空间的片段更加合理。实验表明, 该算法能提高缓存命中率及系统的启动延时, 性能较优。

**关键词:** 流媒体; P2P; 缓存替换算法; 流行度

中图分类号: TP37

文献标识码: A

文章编号: 1674-7720(2011)07-0065-04

## A cache replacement algorithm of media streaming distribution based on P2P

Zhang Jirong, Liu Yanjun

(Department of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

**Abstract:** According to the different characteristics of programme's popularity, the paper proposes a cache replacement algorithm of media streaming distribution based on P2P. All of the video segments are divided into three categories. Before entering the system, different priorities is given and upgraded periodically. And the visited times and frequencies of the segments are considered, so it is more reasonable when the programme block is moved out of cache. The simulation experiments indicate that this algorithm can make the caching system maintain better hit ratio and start delay.

**Key words:** streaming media; P2P; cache replacement algorithm; popularity

随着互联网的飞速发展, 以它为载体应用越来越多样化, 传统应用已经不能充分满足人们的需要, 对通过 Internet 提供更多宽带增值业务的需求已显得越来越迫切。数字多媒体技术的日趋成熟以及网络传输带宽的增加使得在互联网上开展如视频会议、IPTV (Internet Protocol Television)、VoD (Video on Demand) 等各种多媒体应用逐步成为现实。传统多媒体的播放方式是用户先将视频文件下载至本地再进行观看, 而采用流媒体技术只需在播放时将部分内容缓存, 边传送边播放, 节省了下载等待时间和存储空间。但流媒体文件的体积一般都十分庞大, 且对延迟、抖动、包丢失率等较敏感, 会造成用户可感知的服务质量降低。采用高效的流媒体缓存替换策略可以改善这种状况。首先, 从流媒体技术边下载边播放的特点可以看出, 使用缓存技术可弥补网络中的延迟和抖动对视频播放带来的不良影响; 其次从全局看, 缓存可缓解对骨干网络带宽以及中心服务器的压力; 另外, 不管是代理服务器还是客户端的存储空间都

是有限的, 为了充分地利用存储空间, 保障视频文件的流畅播放, 必须依赖于高效的缓存替换策略。

### 1 基于 P2P 的流媒体系统

近年来, P2P (Peer to Peer) 技术在文件共享和网络电话业务中被成功运用, 采用 P2P 技术构建流媒体分发系统成为了比较理想的候选方案。其主要原因在于, 通过这种方式不仅可以获得较好的系统性能和可扩展性, 而且不必改变现有网络结构。最开始 P2P 与流媒体技术结合的成果是 P2P 实时节目直播系统, 从传统的树型分发 (如 ZIGZAG) 到基于 Gossip 的纯 Mesh 分发 (如 Coolstreaming 和 Anysee<sup>[1]</sup>)。在 P2P 方式下, 网络中的每个对等节点 (Peer) 同时是服务提供者和服务享用者, 它们互相协作, 各自贡献自己的一部分资源。然而在 P2P 网络中, 节点存储能力、节点间网络链接带宽的有限性决定了数据无法以稳定的速率连续地在节点间传输, 而在流媒体应用中, 连续且稳定地传输流数据是保证视频播放质量必不可少的条件, 网络的异构性、不可预测的网络抖动使得

这一条件更难以实现。因此,必须采取有效的措施确保一定程度的数据冗余,以利于为节点播放器提供连续的流数据。流媒体缓存技术通过缓存热门视频节目的部分或全部数据,为后来的用户请求提供服务,减少了启动延迟,降低了骨干网络和流媒体服务器负载,而成为了近年来网络应用的研究热点。

## 2 现有流媒体缓存策略

目前存在的缓存策略主要有两类:一类是考虑不同的分段方法来提高资源缓存的效率;另一类是基于资源被访问时的不同特征来设计。

分段方法的讨论集中在如何将大容量的流媒体对象划分成合适的片段,提高网络传输的效率,前缀缓存算法<sup>[2]</sup>有效减小了客户端启动延时;批处理补丁结合前缀缓存思想重点在于提高命中率,减少用户的等待时间;在此基础上提出的最优批处理补丁预先缓存算法<sup>[3]</sup>(BPP)通过补丁的预先缓存,减少了用户的等待时间,节省了主干网络的带宽;指数增长的分段(Exponential Segmentation)策略<sup>[4]</sup>能够快速替换片段来适应缓存对象的访问模式的变化;基于自适应和滞后<sup>[5]</sup>(Adaptive and Lazy)的分段方法根据用户对于不同的流媒体对象的访问频率和访问长度来自适应地改变分段长度和选取删除段。SCU算法<sup>[6]</sup>旨在提高缓存的前缀字节命中率,减少访问延迟,降低流媒体播放时的网络传输成本。

在资源被访问的各种特征中,最有影响的当数资源访问的局部性和资源的被访问次数。LRU(Least Recently Used)和LFU(Least Frequency Used)算法就是分别考虑访问近期性和访问频率的实现方式<sup>[7]</sup>,但LFU存在缓存污染问题,LRU存在长环模式问题。同时,这两种算法还容易出现同一媒体对象的连续替换问题,导致缓存内容被完全释放的概率增大,请求命中率下降和响应延迟增加。LRU-K<sup>[8]</sup>和LCB-K<sup>[9]</sup>考虑了对象最近K次被引用的信息,将访问频率和访问的最近性综合到价值函数的设计之中,具有较好的性能。EELRU<sup>[10]</sup>通过侦测循环访问模式的长度,以自适应选择替换出的对象。还有的是采用将前缀和后缀分开缓存并采用不同替换策略的方式,但这样增加了替换算法开销及存储空间管理的复杂程度。

衡量一个缓存替换策略的主要指标一般有几项:

(1)延时时间(Latency Time):从用户发出一个访问请求开始,到用户在接收到该请求后响应为止,所经历的时间为延时时间,包括启动时和播放过程中的延时。延时时间越短,网络的服务质量越好,同时这也是衡量骨干网能力的一项重要依据。

(2)服务器负载(Server Load):由于用户向服务器发出视频节目请求以及服务器向Peer节点传送数据产生

的负载。在Peer处部署缓存并应用缓存替换策略,通过节点之间的协作可有效降低服务器的负载。

(3)缓存命中率(Cache Hit Rate):缓存命中的次数与用户总的请求数之比,若用 $S_r$ 表示Peer接收到的用户总请求次数, $S_h$ 表示缓存命中次数,则缓存命中率 $CHR=S_h/S_r$ ,命中率越高,系统缓存的效果越好。

(4)空间利用率(Space Use Rate):已经使用的缓存空间大小与缓存总空间的比值,用 $D_u$ 表示已经使用的空间大小, $D_w$ 表示缓存总空间的大小,则空间的利用率 $SUR=D_u/D_w$ ,该值越高,缓存的效率越高。

## 3 缓存替换算法设计

经过对已有缓存算法的研究发现,这些算法都是将所有的视频文件片段一视同仁地处理,即根据视频片段的某些参数来进行替换操作。例如,基于流行度的缓存替换算法<sup>[11]</sup>,流行度参数主要由片段的被访问次数决定,假如一个具有高热度的片段在刚进入系统的一段时间内被访问次数不多,就很有可能被替换出去,这样就造成对热门视频片段的错误操作,从而降低了系统的工作效率,也降低了用户的服务体验。为此,本文提出一种称为PPR(Priority Popularity Recency)的缓存替换算法。该算法同时考虑属于不同类型视频的片段的优先级、片段的流行度、片段的最近一次命中时间这三个因素,尤其是片段的优先级的引入。首先,从视频的受欢迎程度这个角度对它们进行分类,即在中心服务器向下分发视频时,预先给三种不同的视频赋予不同的优先级,使得最受欢迎的视频节目在总的系统中保留的数目最多。然后对存储空间里已有视频片段的流行度进行统计后比较,使得“过期”的片段尽可能地被替换出去。而对于刚刚进入网络并且具有高流行度潜力的新视频片段,为了避免由于其被请求的次数还未来得及累积到一定数值就被替换出存储空间,所以将片段的最近一次命中时间这个因素考虑进来。

### 3.1 PPR 缓存替换算法参数描述

#### 3.1.1 视频的片段的优先级

假设系统中共有 $M$ 个不同的视频节目 $Prog_i(M \geq i \geq 1)$ ,每个节目又被分为 $N_i$ 个片段 $ProgSeg_{i,j}(N_i \geq j \geq 1)$ ,该算法以视频的某一个片段为处理对象,视频的播放速率为540 kb/s,每次处理60 s大约4 MB的数据。

在本文的探讨中将视频分为以下三种:

(1)热门视频:网络中刚刚更新的视频资源,一般是在电视台或电影院发布资源之后的一段时间里面,在网上同时需要观看该类资源的用户很多,如最新电影或最新一期热门节目等。

(2)经典视频:网络中那些一直都会有用户点播观看的经久不衰的视频节目,这类资源的特点是用户对它的请求保持在一个相对比较稳定的水平上,即一直会被访问,但是并发访问量不像访问热门资源那么高,如一些

经典影片或网络课程等。

(3) 冷僻视频: 除去以上两类视频, 还有一类被访问的总次数较低, 并且并发人数也很少的视频, 如很久以前的节目或是受众比较少的资源等。

给此三类视频节目赋予不同的优先级  $Priority_i$ , 对应某类视频的片段优先级表示为  $Priority_{i,j}$ 。热门视频的优先级最高, 其次是经典视频, 冷僻视频的优先级最低。需要强调的一点是, 上述三种视频之间没有绝对的界限, 根据对节点存储空间内的视频进行记录统计, 它们之间存在着互相转化的可能。例如, 热门视频会逐渐变成经典视频或冷僻视频, 冷僻视频也有转化成热门视频的可能性。因此, 本算法不仅仅是简单利用优先级  $Priority_{i,j}$ , 而是每隔时间  $T$  对存储空间里所有片段的优先级进行更新 ( $T$  值通过试验来确定)。

### 3.1.2 视频的片段的流行度

设缓存替换发生的时刻为  $t$ , 则有:  $ProgSet_i(t)$  表示  $t$  时刻存储空间里不同视频组成的集合;  $ProgSegSet_{i,j}(t)$  表示  $t$  时刻存储空间里属于第  $i$  个视频的片段组成的集合;  $ProgReq_i(t)$  表示第  $i$  个视频从开始进驻存储空间到  $t$  时刻被请求的总次数;  $ProgSegReq_{i,j}(t)$  ( $i \in ProgSet_i(t), j \in ProgSegSet_{i,j}(t)$ ) 表示节目  $i$  的第  $j$  段在  $t$  时刻被请求的总次数, 并且有:

$$ProgReq_i(t) = \sum_{j=1}^M ProgSegReq_{i,j}(t) \quad (1)$$

它的流行度为:

$$Pop_{i,j}(t) = ProgSegReq_{i,j}(t) / ProgReq_i(t) \quad (2)$$

### 3.1.3 视频的片段的最近一次命中时间

设该视频片段最近一次的命中时间为  $t_i$ , 则本算法的最后一个参数:

$$Rencency_{i,j} = 1 / (t - t_i) \quad (3)$$

最终, 该视频片段的权值:

$$\varphi_{i,j}(t) = Priority_{i,j} \times Pop_{i,j}(t) \times Rencency_{i,j} \quad (4)$$

## 3.2 PPR 缓存替换算法结构描述

PPR 缓存替换算法由视频优先级更新算法和片段替换算法两个算法构成, 整个算法流程如图 1 所示。视频优先级更新算法是为了确定片段所属视频的优先级。由于视频的受欢迎程度是一个相对较大的时间粒度, 而片段的权值是在一个更小的时间粒度上计算, 如果将视频受欢迎程度放到删除每一个片段上计算, 则会增加不必要的开销。片段替换算法则是当存储空间不足以存放下一个新片段时, 在片段优先级确定的基础上再对其进行评估从而选出淘汰的片段, 完成替换。两个算法的伪代码如下。

#### (1) 视频优先级更新算法

```

for everytime
  for each in Cache
    Record(i) = Segments arrival sequence of Programme(i) in
    
```

certain time

```

Priorityi,j = f(Record(i))
return
return
(2) 视频片段替换算法
if NewSegi,j not in Cache
{
  if R1 > Segi,j
  {
    cache the NewSegi,j
    return
  }
  calculate the value(φi,j(t)) of all old segments
  select the minimum value(φi,j(t)) and delete it
}
if R2 < SegSizei,j
continue
else cache the NewSegi,j
return
    
```

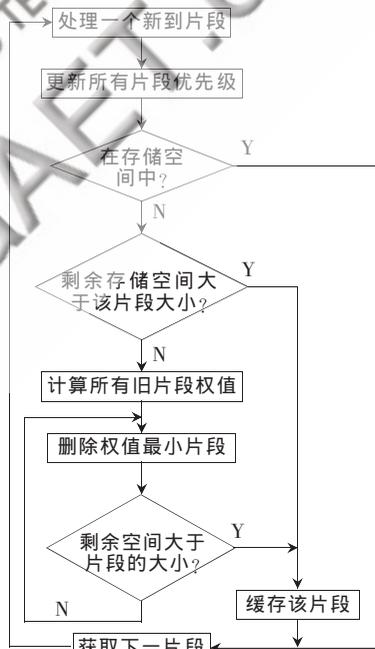


图 1 算法流程图

## 4 实验结果与讨论

为证实前文提出的 PPR 算法要优于常用的缓存替换算法, 本文进行了仿真实验, 实验环境为 VC++6.0。假设客户对 300 个视频片段随机发起请求, 其中 160 个属于热门视频, 100 个属于经典视频, 其余为冷僻视频。现服务器共收到 3 000 个针对不同视频片段的随机请求, 考察替换算法的不同对延迟时间和缓存命中率的影响。本文选择了普通的 LRU、LFU 两个算法作为比较, 进行仿真对比, 实验结果如图 2、图 3 所示。

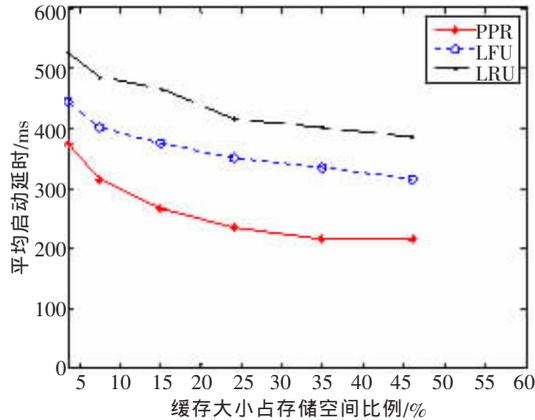


图2 PPR与LRU、LFU的平均启动延时

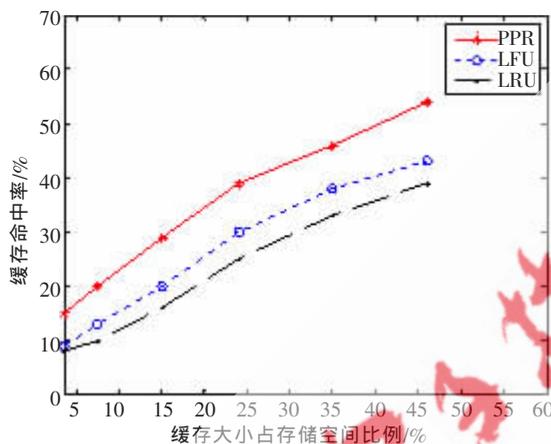


图3 PPR与LRU、LFU的缓存命中率

图2显示了平均启动延迟相对于不同缓存大小的变化情况。可以看出,随着缓存空间的增大,三种算法的启动延迟都呈下降趋势,LFU和LRU算法性能都不如PPR,这主要是由于连续替换使文件最终被全部替换出缓存。而PPR算法预先在内容服务器的存储空间内按受欢迎程度调整三种不同视频片段的比例,使最容易被请求的数据具有较高的权值来解决这个问题,因此在降低系统平均启动延迟方面比较有优势。图3是缓存命中率相对不同缓存大小的变化情况,三种算法的命中率都呈上升趋势,LRU的命中率最低,PPR算法因综合考虑了视频片段的受欢迎程度、被请求的强度以及最近被访问的特征,而且根据实际情况定期更新受欢迎程度这个参数,因此在缓存命中率方面的性能最好。

本文首先研究分析了基于P2P的流媒体系统,然后对现有的流媒体缓存算法进行了对比分析,提出了一种高效的流媒体缓存替换策略。实验证明,本算法在延迟时间和缓存命中率两方面性能更好。另外,将文件大小、传输成本等其他因素引入本算法,并且做进一步的实验对比是下一步的工作。

## 参考文献

- [1] 郑常熠,王新.P2P视频点播内容分发策略[J].软件学报,2007,18(11):2942-2954.
- [2] SEN S, REXFORD J, TOWS L. Proxy prefix caching for multimedia streams [C]. Proceedings of IEEE Infocom, New York, 1999: 1310-1319.
- [3] RIZZO L, VICISANO L. Replacement policies for a proxy cache [J]. IEEE/ACM Transactions on Networking, 2000,8(2):158-170.
- [4] WU K L, YU P S, WOLF J L. Segment-based proxy caching of multimedia streams [C]. Proceedings of the 10th International Conference on World Wide Web, WWW'01, Hongkong, China, 2001:56-60.
- [5] CHEN S, SHEN B, WEE S, et al. Adaptive and lazy segmentation based proxy caching for streaming media delivery [C]. Proceedings of ACN NOSSDAV. Monterey, CA, 2003:694-703.
- [6] LIM E, PARK S H, HONG H O, et al. A proxy caching scheme for continuous media streams on the Internet [C]. The 15th International Conference on Information Networking. Beppu City, Oita, Japan, 2001:720-725.
- [7] KRISHNAMURTY B, REXFORD J. Web protocol sand practice: HTTP/1.1, networking protocols, caching and traffic measurement [M]. Addison Wesley Longman Publishing Co., 2001.
- [8] O'Neil E J, O'Neil P E, WEIKUM G. An optimality proof of the LRU-K page replacement algorithm[J]. Journal of the ACM, 1999, 46 (1):92-112.
- [9] OTOO E, OLKEN F, SHOSHANI A. Disk cache replacement algorithm for storage resource managers in data grids [C]. Proceedings of IEEE / ACM Conference on Supercomputing, Baltimore, Maryland, USA, 2002:1-15.
- [10] SMARAGDAKIS Y, KAPLAN S, WILSON P. The EELRU adaptive replacement algorithm [J]. Elsevier Science Performance Evaluation, 2003,53(2):93-123.
- [11] 杨传栋,余镇危.基于流行度预测的流媒体代理缓存替换算法[J].计算机工程,2007,33(7):99-100.

(收稿日期:2010-12-03)

## 作者简介:

张继荣,女,1963年生,教授,博士,主要研究方向:宽带通信网。

刘艳君,女,1986年生,硕士研究生,主要研究方向:通信网,流媒体技术。