

基于媒体流行度和前缀缓存的缓存替换算法

刘宜宁 赵正德 全卫新 张伟

(上海大学计算机科学与工程学院, 上海 200072)

摘要 针对流媒体用户访问偏好的情况,提出了一种基于前缀缓存与媒体流行度的缓存替换算法。该算法根据不同媒体外部、内部流行度给出预估的综合流行度,进而选择可用缓存中具有低流行度的片断进行替换,使得缓存中的所有片断的再利用价值之和最大。模拟实验结果表明,该算法能减少缓存的替换次数,提高缓存命中率,性能较好。

关键词 流媒体 代理缓存 流行度 前缀缓存

中图分类号: TP393 **文献标识码**: A **文章编号**: 1006-8961(2007)10-1753-04

Proxy Cache Replacement Algorithm Based on Popularity and Prefix Caching

LIU Yi-ning, ZHAO Zhen-de, QUAN Wei-xin, ZHANG Wei

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072)

Abstract Considering user access preference in streaming media, the paper proposes a new proxy caching replacement algorithm based on prefix caching and media popularity. The algorithm calculated the predicted popularity using the external and internal popularity of media segment. Then the segment with low popularity is chosen to be replaced. Simulation results demonstrate that, the method can achieve high caching hit rate and reduce request delay under same conditions.

Keywords streaming media, caching proxy, popularity, prefix caching

1 引言

随着近年来社区宽带网的高速发展,各种形式的流媒体服务迅速发展,如视频点播、IPTV(internet protocol television)、播客(podcast)等。随之而来的问题是,由于互联网上用户的激增,造成了骨干传输网络带宽紧张,并且增加了用户接受服务所经历的时延,大大地降低了服务质量,严重影响了用户的观看。解决该问题的一个普遍方法是在网络边缘靠近用户的地方部署代理缓存服务器,代替流服务器为用户提供服务。图1描述了典型的基于代理缓存的流媒体分发服务结构。

相对于现在已广泛使用的用于传统Web对象的代理缓存,流媒体代理缓存有如下特点^[1]:

(1) 流媒体对象往往要比传统Web对象文件大得多。一般的Web对象以KB为计量单位,而流媒体对象多以MB甚至GB作为单位。因此,缓存只能存储少量完整的流媒体文件。

(2) 经过统计发现,用户通常只浏览流媒体的最初部分以决定是否全部观看,媒体各时间段的受欢迎程度有随着播放时间的增加而递减的过程。

(3) 根据用户对流媒体对象的访问行为知道,在一个流媒体服务系统中,80%的用户请求访问的是20%的媒体对象,也就是说大量的用户请求都集中在少量比较流行的媒体对象上。

(4) 流媒体传输对网络带宽、时延、丢包率等有较高的要求,这些因素都会直接影响到用户观看节目的体验。

由于以上特性的存在,所以需要采取一种不同

收稿日期:2007-07-16;改回日期:2007-07-25

第一作者简介:刘宜宁(1982-),男,现为上海大学计算机应用专业硕士研究生。主要研究方向为计算机网络、多媒体技术。E-mail: liuyining_82@hotmail.com

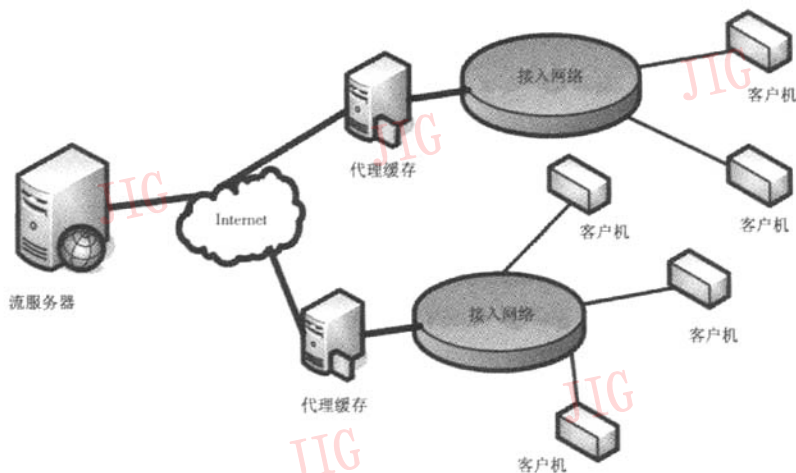


图 1 基于代理缓存的流媒体分发服务系统结构

Fig. 1 Structure of streaming media delivery system based on proxy cache

于 Web 对象缓存的流媒体缓存技术。

目前已有的缓存策略大体可分为两类：对象备份和部分缓存。对象备份是将媒体对象从流服务器整个复制给边缘服务器（edge server proxy, ESP），将单个文件对象作为一个整体缓存。部分缓存针对流媒体文件占用缓存空间大的特点，对每一文件只缓存其部分内容，比较典型的有前缀缓存（prefix caching）和分段缓存（segment caching）以及基于流行度缓存（popularity caching）。其中前缀缓存可以降低或消除客户端的启动延迟，而基于流行度缓存可以有效地提高缓存字节命中率，减轻骨干网压力。不管是以上哪种策略，代理服务器的缓存替换算法都是研究的重点。由于代理缓存服务器上缓存的容量有限，要保证缓存代理服务器能充分发挥其作用，必须要缓存那些最有用的部分，使得缓存中所有对象的再利用价值之和最大。

2 基于媒体流行度和前缀缓存的缓存替换算法

为简单起见，假定：

(1) ESP 到用户侧有丰富的带宽供应（一般的小区局域网带宽可达 10 ~ 100M）。

(2) ESP 具有足够的存储容量存储流服务器上所有流文件的头部（如前 20s）和部分作为代理缓存所需的空。

(3) 从 ESP 到用户的传输延迟远小于从流服务器到 ESP 的传输延迟。

算法结合前缀缓存和流行度缓存的优点，使得用户浏览流媒体的相应时间得到缩短并且缓存命中率得以提高。算法中，在 ESP 上缓存所有流文件的开始部分，当 ESP 上缓存空间不足需要替换时，则根据流行度系数替换出预估的利用价值最小的分块。

2.1 媒体流行度的定义与计算

流行度是用于描述流媒体流行程度的一个概念，它反映了流媒体文件在某一时刻被用户请求的概率。其中又可以细分为外部流行度和内部流行度。

外部流行度是各个流媒体间的流行系数，研究表明典型的流媒体流行度模型遵循 Zipf 规律分布，但它并没有直接给出 N 个流媒体对象中在最近一段时间内哪个最流行，为此采用以下方法来描述外部流行度。

设 $R(i)$ 表示最近 K 次流媒体请求中对流媒体文件 i 的请求次数，流媒体文件 i 的外部流行度定义为

$$P_e(i) = R(i)/K \quad (1)$$

K 的取值大小决定了统计宽度。

内部流行度^[2]是指媒体访问者在不同时刻中止对媒体对象的浏览，使得流媒体内部不同片段具有不同的流行度，例如，影片起始部分往往具有更高的流行度。

这里采用文献[3]中提出的指数增长分段策略

作为对流媒体分段的依据,将基本片段长度 L_b 取为与前缀长度等同,即

$$L_b = L_p \quad (2)$$

该策略可以有效降低用户请求阻塞的概率,为了适应用户访问流行度变化,如图 2 所示,后面的片段长度呈指数增长,第 i 个片段的长度则为

$$L_i = 2^{i-1} L_b \quad (3)$$



图 2 基于指数增长的分段策略

Fig. 2 Segment strategy based on index increasing

对于一个缓存在 ESP 上的流媒体对象片段,用访问度来表示该流媒体在 ESP 上的流行系数,访问度是用户观看流媒体片段的次数与观看该流媒体次数的比率。则流媒体 i 的 j 片段的访问度为

$$A_p(i, j) = \frac{C(i, j)}{C(i, 1)} \quad (4)$$

其中, $C(i, j)$ 表示对于流媒体 i 的 j 片段被访问的次数,可以看出,由于用户在媒体播放过程中随时会中止, $C(i, j) \leq C(i, j-1)$ 。

由式(1)、式(4)给出流行度系数 P :

$$P(i, j) = P_s(i) A_p(i, j) = \frac{R(i) C(i, j)}{K \cdot C(i, 1)} \quad (5)$$

由于流行度具有随时间变化的特性,因此一个流媒体文件在未来某一时刻的流行度是无法预知的,为此根据其历史流行情况进行预估,这里采用类似于文献[4]中的衰减函数法,该方法通过如下函数来计算:

$$f_{k+1}(j) = f_k(j) * 2^{-c/K} \quad (6)$$

式中, $f_{k+1}(j)$ 表示对缓存中片段 j 的第 $k+1$ 次访问时 j 的流行度估计值, c 表示从第 k 次访问到第 $k+1$ 次访问之间,用户访问其他文件的总次数, K 为常量,用于控制衰减速度。

2.2 算法的基本思路

考虑了文件最近 K 次访问情况,使文件的前缀部分拥有最高的缓存价值,提高了用户的响应延迟,减少文件被连续替换情况的发生。

算法约定如下:

(1) $f(i, 1) = 1$; ($i = 1, 2, 3, \dots$, 媒体总数) // 每个流媒体的前缀部分具有最高的流行度系数;

(2) $Remain$ 为当前剩余的缓存空间大小;

(3) $Size(i, j)$ 为缓存文件 i 的 j 片段大小。

算法的伪代码描述如下:

CachingReplace(待缓存文件 i 的 j 片段)

```

{
if (Remain > Size(i, j)) // 剩余空间足够缓存待缓存片段
{
缓存文件  $i$  的待缓存段  $Size(i, j)$ ;
Remain - = Size(i, j);
return;
}
while (Remain < Size(i, j)) {
while(缓存中除了属于  $i$  文件之外的其他缓存片段集合  $K$ ) {
计算  $K$  中每个片段的预估流行度;
取  $K$  中具有最小流行度的  $m$  文件的  $n$  片段,记其为  $(m, n)$ ;
if( $n$  片段预估流行度大于  $j$  片段的预估流行度) {
return;
}
Remain + = Size( $m, n$ );
}
缓存  $j$  片段;
Remain - = Size(i, j);
return;
}
}
    
```

3 实验结果分析

针对提出的基于流行度的前缀缓存算法建立了原型系统,并对其性能与其他内容分发系统进行了比较,结果如表 1 所示,其中将实际数据中访问量靠前的 115 部影片的用户行为作为仿真数据的输入,视频流媒体按照 10s 播放长度为前缀大小的指数增长分段。

表 1 采用基于流行度前缀缓存算法的系统与现有系统比较

Tab. 1 Comparisons of two systems

评价指标	采用基于流行度前缀缓存系统	主流内容分发系统
客户端请求延迟(s)	<1	<3
带宽降低率(%)	60	20
ESP 命中率(%)	80	30
并发提供节目流	250 ~ 300	200

从实验结果可以得出,采用基于流行度的前缀替换算法的系统,可以提高客户端对媒体请求的平

均响应速度,避免了用户在过去请求未缓存媒体时发生的延迟过于大的情况发生,缓存命中率指标的提高明显。而带宽降低率较缓存命中率稍有不足,原因是在指数增长分段策略中,随着服务时间增大,单个分块传输成本也将成倍增长,在以后的研究中可以采用算法复杂度更高的平均分段策略来解决。

4 结 论

随着网络流媒体需要的持续增加,人们对于流媒体服务质量的要求越来越高。代理缓存技术是提高流媒体分送系统效率、降低骨干网络带宽消耗的重要手段。本文提出了一种新的基于流媒体流行度的前缀缓存替换算法,实验中验证了该算法具有较好的性能,在有效地降低用户等待时间的同时,提高了缓存命中率。以后工作重点放在将流媒体字节有用性、文件分段、流媒体文件码率等因素分别引入算法中,以便进一步提高系统性能,从而提高系统对于

不同网络环境的适应性。

参考文献 (References)

- 1 Tian Xiao-bo, Chen Shu-yu. Proxy cache replacement algorithms for streaming media based on smallest cache utility [J]. Journal of Computer Applications, 2007, 27(3): 733 ~ 736. [田小波,陈蜀宇. 基于最小效用的流媒体缓存替换算法[J]. 计算机应用, 2007, 27(3): 733 ~ 736.]
- 2 Yu Jiang, ChunTung Chou, Yang Zhong-kai, et al. A proxy caching algorithm based on user access preference in streaming media [J]. Microelectronics & Computer, 2006, 23(11): 19 ~ 22, 25. [余江, ChunTung Chou, 杨宗凯等. 基于媒体用户访问行为偏好模型的代理缓存算法[J]. 微电子学与计算机, 2006, 23(11): 19 ~ 22, 25.]
- 3 Wu K, Yu P, Wolf J. Segment-based proxy caching of multimedia streams[A]. In: Proceedings of IEEE INFOCOM[C], Hongkong, China, 2001: 36 ~ 44.
- 4 Jin S, Bestavros A. Popularity-aware Greedy dual-size web proxy caching algorithms [A]. In: Proceedings of ICDCS [C], San Francisco, California, USA, 2000: 254 ~ 261.