

文章编号:1001-9081(2007)03-0733-04

基于最小效用的流媒体缓存替换算法

田小波¹, 陈蜀宇²

(1. 重庆大学 计算机学院, 重庆 400044; 2. 重庆大学 软件学院, 重庆 400044)

(xbtian69@163.com)

摘要: 提出最小缓存替换算法 SCU-K, 综合考虑流媒体文件最近 K 次访问情况, 使缓存大小动态适应媒体流行度、字节有用性和已缓存部分大小的变化, 降低了文件前缀部分被替换的概率, 避免 LRU 和 LFU 算法中出现的媒体文件被连续替换的问题。在与 LRU, LFU 和 LRU-2 算法的对比实验中, SCU-K 算法在提高缓存空间利用率、字节命中率和降低启动延迟方面具有更好的性能。

关键词: 流媒体; 代理缓存; 缓存替换算法; 字节有用性

中图分类号: TP37 文献标识码: A

Proxy cache replacement algorithms for streaming media based on smallest cache utility

TIAN Xiao-bo¹, CHEN Shu-yu²

(1. College of Computer Science, Chongqing University, Chongqing 400044, China;

2. School of Software Engineering, Chongqing University, Chongqing 400044, China)

Abstract: The Smallest Cache Utility-K (SCU-K) algorithm was proposed. It comprehensively considered the latest K times of visits in order to make the cached size dynamically adapt to the change of the media popularity, bytes benefit and the size of cached parts. In such cases, the probability of the file's prefix part being replaced was reduced and such problem of the LRU and LFU algorithm as the streaming media file being continuously replaced and finally being released was avoided. The experiment proves that the SCU-K algorithm has better performance in increasing cache space utilization rate and byte hit ratio, and reducing startup delay than LRU, LFU and LRU-2.

Key words: streaming media; proxy cache; cache replacement algorithm; bytes benefit

Internet 的飞速发展使其成为了社会基础信息的载体。网络上的文本和图片等静态信息已经不能充分满足人们的需要。日益成熟的数字多媒体技术使得在 Internet 上开展各种流媒体应用如 VoD(Video on Demand)、IPTV(Internet Protocol Television)、远程教育、视频会议等已逐步成为现实。然而, 由于现有的互联网络是建立在传输非实时数据的基础上的, 缺乏对延迟、抖动、包丢失率等敏感的实时通信的良好支持, 影响了流媒体传输的性能, 造成用户可感知质量的降低。流媒体代理缓存技术^[1] 通过在骨干网络的边缘靠近用户的地方部署代理服务器来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。

1 流媒体代理缓存的特点

流媒体代理缓存的系统结构如图 1 所示, 其相对于现在已广泛使用于 Web 对象的代理缓存技术有如下特点:

1) 流媒体对象的大小通常比传统 Web 对象要大几个数量级。因此, 完全缓存对象的实现方式在流媒体代理缓存中会大大降低缓存空间利用率和命中率, 并不现实。

2) 流媒体对象通常都具有一次写多次读的性质, 很少进

行改变, 因此在流媒体缓存中, 缓存的一致性并不是非常重要的问题。

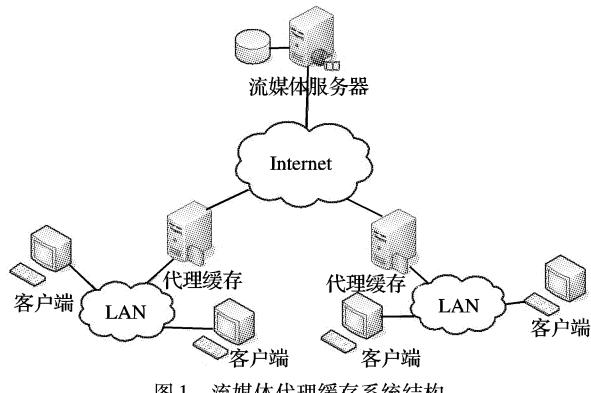


图 1 流媒体代理缓存系统结构

3) 用户通常只浏览流媒体文件的起初部分以决定是否继续观看, 因此在设计流媒体代理缓存系统时必须考虑如何适应该情况。

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质量。

这些特性的存在, 使得不能直接照搬传统的 Web 缓存技术, 而必须开发一种适用于流媒体的代理缓存技术。

收稿日期: 2006-09-04; 修订日期: 2006-11-27 基金项目: 教育部新世纪优秀人才支持计划项目(NCET-04-0843)

作者简介: 田小波(1981-), 男, 重庆人, 硕士研究生, 主要研究方向: 多媒体网络技术、网络管理与应用; 陈蜀宇(1963-), 男, 重庆人, 教授, 博士, 主要研究方向: 操作系统、多媒体网络技术、网络通信。

2 相关工作

在流媒体代理缓存的设计中,现有的缓存算法研究几乎都是基于部分缓存进行的,主要差别体现在分段策略和性能影响因素上。分段策略方面,Sen 等人^[2]提出前缀缓存算法以减少流媒体的延迟和抖动。Lim 等人^[3]提出 SCU 算法,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。Wu 等人^[4]提出指数增长的分段策略能够快速替换片段来适应缓存对象的访问模式的变化。Chen 等^[5]提出一种基于自适应和延迟的分段方法。在缓存性能影响因素方面,最有影响的当数资源访问的局部性原理^[6]和资源的访问频率^[7]。LRU 和 LFU 算法就是分别考虑访问近期性和访问频率的实现方式,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持续替换同一媒体对象的问题,导致文件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。鉴于此,EELRU^[8,9]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。文献[4,10]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。LRU-K^[11,12]和 LCB-K^[6]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。RBC 算法考虑了文件大小和所需的发送带宽因素,ATCB-SCU 算法则重点考虑传输代价。Interval Caching 策略^[13]通过结合磁盘缓存和内存缓存技术提高了性能,但它的有效性随着访问间隔的增加而失效。LRFU 统一考虑访问频率和近期性,在两者之间进行折中。现有的缓存替换算法在建立缓存价值函数时,几乎没有考虑缓存的字节有效性,而缓存字节有效性^[14]对提高缓存空间利用率和字节命中率具有重要的影响,但文献[14]中的算法采用的是整体缓存的思想,具有一定的局限性。

3 算法设计

3.1 算法模型

在保证缓存内容能有效屏蔽代理服务器获取媒体剩余部分内容延迟的情况下,将媒体文件等分成若干段,正在对请求进行服务的流媒体文件缓存内容不允许被替换。设缓存替代请求时刻为 t ,记此时缓存中的流媒体文件集合为 $FileSet(t) = \{1, 2, \dots, n\}$,文件 i 在代理服务器中缓存部分的大小为 $Size_i(i \in FileSet(t))$;估计文件 i 以后会被使用的概率为 $P_i(t)$,缓存容量上限为 B 。为了最大化缓存空间的利用率和字节命中率, t 时刻执行缓存替代策略后应使得:

$$\text{Maximize} \sum_{i \in FileSet(t)} P_i(t) \times \delta_i(t) \quad (1)$$

同时满足约束条件:

$$\sum_{i \in FileSet(t)} Size_i \leq B \quad (2)$$

其中, $\delta_i(t)$ 表示媒体文件的字节有用性,由于用户在请求某个媒体文件服务时,可能在任意播放位置停止。因此,媒体文件对于用户来说,并不一定是整体有用,而是部分地有用,应该对文件 i 中具有使用价值的部分进行评估,即考虑媒体文件的字节有用性。设文件 i 大小为 s_i ,共播放 m 次,第 j 次

播放时间为 $t_j(0 < j \leq m)$,平均播放速率为 r_i ,则对 $\delta_i(t)$ 的计算如下:

$$\delta_i(t) = \frac{\sum_{j=1}^m r_i \times t_j}{s_i} \quad (3)$$

对于(2)式,假定每个文件的缓存大小远小于总的缓存空间大小,因此缓存文件占用的空间达到最大时,可以忽略剩下的部分空间,则(2)式可近似为:

$$\sum_{i \in FileSet(t)} Size_i = B \quad (4)$$

该问题实际上是一个背包问题,求解该问题是 NP 难的。因此,采用近似最优化算法^[15],对每个文件 i ,在时刻 t 建立缓存效用函数:

$$\varphi_i(t) = \frac{P_i(t) \times \delta_i(t)}{Size_i} \quad (5)$$

在算法执行时,计算每个媒体文件的缓存效用值,每次替换出效用值最小的文件的最后一段。在效用函数中,综合考虑媒体文件最近 K 次访问的情况和在缓存中的缓存大小,结合媒体文件的字节有用性,动态地改变文件的缓存效用。随着文件缓存部分的减小,其缓存效用会逐渐变大,使得同一文件中越靠近起始部分的段拥有相对越高的缓存效用,从而避免了在 LRU 和 LFU 替换算法中出现文件被连续替换,导致文件缓存内容被完全释放,缓存空间利用率下降和播放启动延迟增大等问题。同时,算法考虑了文件的字节有用性对缓存空间利用率的影响。

3.2 访问概率估算

在式(5)建立的缓存效用函数中,参数 $\delta_i(t)$ 和 $Size_i$ 都比较容易计算,关键是对参数 $P_i(t)$ 的估算。由于对任意文件 i 的访问假定独立且服从参数为 $\lambda_i(t)$ 的泊松过程通常是有效的,所以可以通过式(6)估算 $P_i(t)$ ^[6]:

$$P_i(t) = \frac{\lambda_i(t)}{\sum_{j=1}^n \lambda_j(t)} \quad (6)$$

因为我们的替换是通过对文件的效用值排序来决定的,因此将(6)式带入(5)中,可简化为:

$$\varphi_i(t) = \lambda_i(t) \times \frac{\delta_i(t)}{Size_i} \quad (7)$$

对于 $\lambda_i(t)$ 的估算,参照文献[6]的估算方法,记文件被评估的最近访问次数为 $k_i(t)$,其最大值为 K , t_{ki} 为文件倒数第 k_i 次访问的时间,则:

$$\lambda_i(t) = \frac{k_i(t)}{t - t_{ki}} \quad (8)$$

将(8)式带入(7)中,得到效用函数为:

$$\varphi_i(t) = \frac{k_i(t)}{t - t_{ki}} \times \frac{\delta_i(t)}{Size_i} \quad (9)$$

3.3 替换算法描述

SCU-K 算法考虑了文件最近 K 次访问情况,使文件的缓存长度随文件被请求概率的变化和文件在多次播放中体现出的字节有用性而动态变化,同时使文件的前缀部分拥有更高的缓存价值,降低文件被连续替换而最终被完全替换出缓存的概率。在算法的描述中,作如下约定:

S_i —文件 i 的缓存段的大小;

R —当前剩余的缓存空间大小;
 W_i —文件 i 的待缓存段;
 L_i —文件 i 的总大小;
 $Size_i$ —文件 i 已缓存部分的大小;
 φ_i —文件 i 的缓存效用。

算法的伪代码描述如下:

```

cachingProcess(待缓存文件 i){
    if ( $L_i > Size_i$ ) { // 文件 i 还没有被完全缓存
        if ( $R > S_i$ ) { // 当前剩余空间足够缓存文件 i 的一个段
            缓存文件 i 的待缓存段  $W_i$ ;
            return;
        }
        计算文件 i 如果缓存下一个段后的缓存效用  $\varphi_i(Size_i + S_i)$ ;
        while(当前缓存中的除 i 外的其他文件 j) {
            计算文件 j 的缓存效用  $\varphi_j$ ;
        }
        选取除 i 外其他文件缓存效用中的最小值  $\varphi_m$ , 记此文件为 k;
        if ( $\varphi_m \geq \varphi_i(Size_i + S_i)$ )
            return;
        从缓存中清除文件 k 的最后一段;
        缓存文件 i 的待缓存段;
        return;
    }
}

```

4 测试与分析

4.1 模拟测试环境

我们设计了一个离散事件仿真器,用以模拟代理服务器对缓存的操作行为,模拟器由用户的请求驱动运行。当有新的请求到达仿真器时,检查缓存是否命中,同时,仿真器按照缓存替换算法决定是否增加缓存和进行缓存替换。模拟器设计和运行参数如表1。

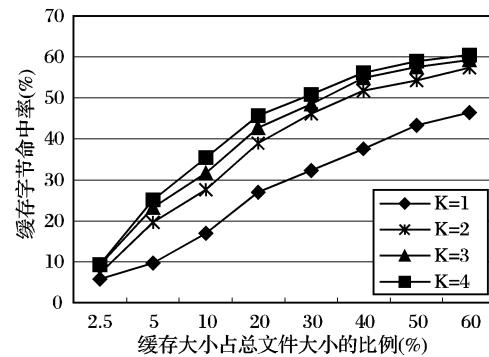
表1 模拟参数说明

| 参数名称 | 参数大小 |
|-----------------|------------------------------|
| 媒体文件大小范围 | 100MB ~ 300MB |
| 媒体文件数量 | 200 |
| 平均请求时间间隔 | 10s |
| 媒体文件流行度 | 服从 $\theta = 0.27$ 的 Zipf 分布 |
| 模拟测试时间 | 模拟 2000 个用户请求 |
| 代理服务器与用户间网络延迟 | 100ms |
| 代理服务器与源服务器间网络延迟 | 200ms |
| 文件分段大小 | 2MB |

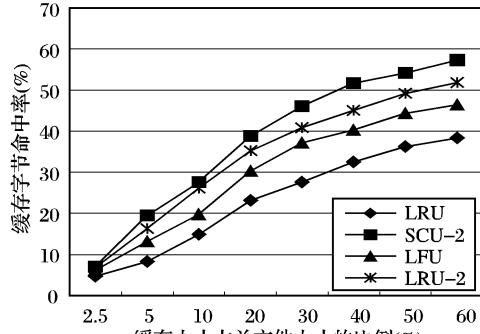
4.2 模拟测试结果及分析

我们对 SCU-K 与 LRU, LFU, LRU-K 算法在缓存字节命中率和平均启动延迟方面进行了对比模拟实验。对于 LRU-K 算法,因为其整体性能在 $K = 2$ 时最好^[11,12],所以选择与 LRU-2 进行对比实验。首先,比较 SCU-K 算法在 K 取不同值时的性能,结果如图 2(a)所示,随着 K 值的增大,SCU-K 算法在字节命中率指标上都呈上升趋势,尤其是当 $K = 2$ 时,性能增幅最明显(增幅最大约为 14.1%),因此对比实验时 SCU-K 中取 $K = 2$ 。图 2(b)显示了缓存字节命中率相对于不同缓存大小的变化情况,随着缓存空间的增大,四种算法在字节命中率上呈上升趋势,LRU 字节命中率最低,LFU 优于 LRU,LRU-

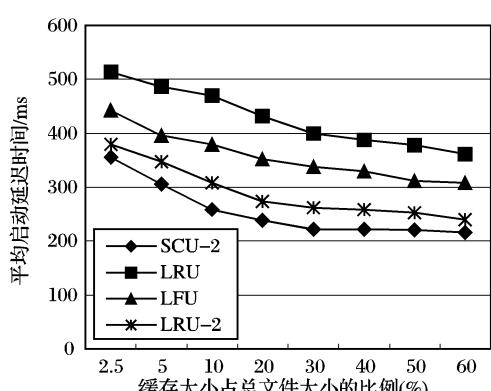
2 比 LFU 的字节命中率增加约 0.7% 到 6.3%,SCU-2 除了考虑文件的访问频率和访问近期性以外,还体现了文件的字节有用性对缓存的影响,增大了缓存空间利用率,在缓存字节命中率方面性能最好,相对于 LRU-2 平均增幅约为 3.9%。图 2(b)显示了平均启动延迟相对于不同缓存大小的变化情况,随着缓存空间的增大,四种算法的启动延迟都成下降趋势,LFU 和 LRU 算法性能明显不如 LRU-2 和 SCU-2,其主要原因是存在连续替换而导致文件最终被全部替换出缓存的问题。SCU-K 通过将媒体文件缓存大小作为缓存效用的变量,使媒体段越靠近起始部分具有相对越高的缓存效用来解决这一问题,降低了文件前缀部分被替换的概率,因此在降低启动延迟方面具有更高的优势。



(a) SCU-K 在不同 K 值下的字节命中率



(b) SCU-2 与 LRU, LFU, LRU-2 的字节命中率



(c) SCU-2 与 LRU, LFU, LRU-2 的平均启动延迟

图 2 SCU-2 与几种常用缓存算法的对比

5 结语

综上所述,流媒体缓存技术是提高流媒体传输系统效率,降低骨干网络带宽消耗,服务器负载和播放启动延迟的重要手段。文中提出的 SCU-K 算法综合考虑了流媒体文件字节有用性对提高缓存空间利用率的影响和前缀缓存对降低启

动延迟的影响,以文件最近 K 次访问情况为基础建立缓存效用作为替换依据,每次替换缓存效用最小的文件的媒体段。SCU-K 算法可以根据文件的访问情况动态调整文件的缓存大小,让文件的缓存大小适应媒体流行度、字节有用性和已缓存部分大小的变化,使越靠近起始部分的媒体段具有相对越高的缓存效用,降低了文件前缀部分被替换的概率,避免 LRU 和 LFU 算法中出现的媒体文件被连续替换而最终被完全替换出缓存的问题。在与 LRU,LFU,LRU-K 算法的对比实验中,SCU-K 算法在提高空间利用率,缓存字节命中率和降低启动延迟方面具有更大的优势。

参考文献:

- [1] 钟玉琢,向哲,沈洪,等.流媒体和视频服务器[M].北京:清华大学出版社,2003. 117-158.
- [2] SEN S, REXFORD J, TOWSLEY D. Proxy prefix caching for multimedia streams[A]. Proceedings of IEEE Infocom[C]. New York, 1999. 1310-1319.
- [3] LIM E, PARK SH, HONG HO, et al. A proxy caching scheme for continuous media streams on the Internet[A]. The 15th International Conference on Information Networking (ICOIN'01)[C]. Beppu City, Oita, Japan, 2001. 720-725.
- [4] WU K, YU PS, WOLF J. Segment-based proxy caching of multimedia streams[A]. Proceedings of WWW[C]. 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[A]. Proceedings of ACM NOSSDAV[C]. Monterey, CA, 2003. 694-703.
- [6] OTOO E, OLKEN F, SHOSHANI A. Disk cache replacement algorithm for storage resource managers in data grids[A]. Proceedings of IEEE / ACM Conference on Supercomputing [C] . Baltimore , Maryland, USA, 2002. 1 - 15.
- [7] ROBINSON JT, DEVARAKONDA MV. Data cache management using frequency-based replacement[A]. Proceedings of SIGMETRIC on Measuring and Modeling of Computer Systems[C]. Boulder, Colorado, USA, 1990. 134 - 142.
- [8] SMARAGDAKIS Y, KAPLAN S, WILSON P. The EELRU adaptive replacement algorithm[J]. Elsevier Science Performance Evaluation, 2003, 53(2): 93 - 123.
- [9] SMARAGDAKIS Y, KAPLAN S, WILSON P. EELRU: Simple and effective adaptive page replacement[A]. Proceedings of SIGMETRICS on the Measurement and Modeling of Computer Systems[C]. Atlanta, Georgia, USA, 1999. 122 - 133.
- [10] 罗彬,王宏远,盛东晖,等.代理服务器视频对象缓存方法的研究[J].武汉大学学报. 2005, 38(5): 141 - 145.
- [11] NEIL EJO, NEIL PEO, WEIKUM G. The LRU-K page replacement algorithm for database disk buffering[A]. Proceedings of International Conference on Management of Data[C]. Washington, DC, USA, 1993. 297 - 306.
- [12] NEIL EJO, NEIL PEO, WEIKUM G. An optimality proof of the LRU-K page replacement algorithm [J]. Journal of the ACM, 1999, 46(1): 92 - 112.
- [13] DAN A, SITARAM D. A generalized interval caching policy for mixed interactive and long video environments[A]. Proceedings of MMCN[C]. San Jose, CA, USA, 1996. 344 - 351.
- [14] 肖明忠,李晓明,刘翰宇,等.基于流媒体文件字节有用性的代理服务器缓存替代策略[J].计算机学报,2004, 27(12): 1633 - 1641.
- [15] SCHEUERMANN P, SHIN J, VINGRALEK R. WATCHMAN: A data warehouse intelligent cache manager [A]. Proceedings of VLDB[C]. Bombay, India, 1996. 51 - 62.

(上接第 732 页)

由表 2 可以看出,MVPSO-NN 模型的拟合性能优于 BPNN 模型。

用没有参加训练的第 10 批数据对两种模型进行检验,结果如图 3 所示,统计结果如表 3 所示。

表 3 两种模型泛化性能比较

| 性能指标 | MVPSO-NN | BPNN |
|-------------------------------|----------|--------|
| $ e < 0.01(\text{g/L})$ 所占比例 | 66.3% | 56.2% |
| $ e < 0.02(\text{g/L})$ 所占比例 | 88.2% | 85.2% |
| 均方差 | 0.0065 | 0.0289 |
| 绝对误差平均值 | 0.0058 | 0.0218 |

由表 3 及图 3 可以看出,两种模型的泛化性能较好,具有较高的预测精度,并且 MVPSO-NN 模型优于 BPNN 模型。

6 结语

多速粒子群算法(MVPSO)是一种改进的 PSO,优化效率和优化性能较标准 PSO 有明显提高。本文将 MVPSO 算法应用于前向神经网络权值和阈值的优化,试验结果显示,MVPSO 算法陷入局部极小值的概率显著降低,比 BP 算法更容易找到全局最优解,而且在应用上灵活方便,目标函数与传递函数没有必须可导的限制,应用范围大为拓宽。将基于 MVPSO 算法的神经网络应用于青霉素发酵过程产物(青霉素)浓度软测量,仿真实验表明,该模型训练误差小、学习速

度快、泛化能力强,可以满足青霉素发酵产物浓度预估要求,为实现发酵过程优化控制提供了有利依据。

参考文献:

- [1] KENNEDY J, EBERHART RC. Particle swarm optimization[A]. Proceedings of IEEE International Conference on Neural Networks [C]. Perth, 1995. 1942 - 1948.
- [2] EBERHART RC, KENNEDY J. A new optimizer using particle swarm theory[A]. Proceedings of the 6th International Symposium on Micro Machine and Human Science[C]. Nagoya, 1995. 39 - 43.
- [3] VAN DEN BERGH F. An Analysis of Particle Swarm Optimizers [D]. Department of Natural and Agricultural Science, University of Pretoria, Pretoria, South Africa, 2001.
- [4] YANG W. Survey on Particle Swarm Optimization Algorithm[J]. Engineering Science, 2004, 6(5): 87 - 94.
- [5] XIE XF, ZHANG WJ. A Dissipative Particle Swarm Optimization [A]. Proceedings of the Congress on Evolutionary Computation[C]. Honolulu, USA, 2002, 2. 1456 - 1461.
- [6] WANG B, DONG YL, WANG CL. A New Particle Optimizer Algorithm and Application [A]. 1 st International Symposium on Systems and Control in Aerospace and Astronautics[C]. 2006, 1. 312 - 315.
- [7] 焦李成.神经网络系统理论[M].西安:西安电子科技大学出版社,1988.
- [8] 常玉清,邹伟,王福利.基于支持向量机的软测量方法研究[J].控制与决策,2005,20(11):1307 - 1310.