

基于混合模式的流媒体缓存调度算法

叶剑虹^{1,2} 叶双^{1,3}

(华侨大学计算机科学与技术学院 厦门 361021)¹ (西安电子科技大学机电工程学院 西安 710126)²
(南京理工大学计算机科学与技术学院 南京 210094)³

摘要 介绍了一种结合了 CDN 和 P2P 互补优势的流媒体混合内容分发网络(HyCDN)。针对 HyCDN 不同区域提出了相应的缓存算法,域内用户端综合考虑了流媒体前缀字节的有用性、文件的传输代价及点播热度,在此基础上提出缓存替换算法(Comprehensive Value Cache Replacement Algorithm for P2P, CVCR4P2P);对域间边缘服务器采用补丁预取与调度算法(Dynamic Scheduling Algorithm for Proxy Caching, DSA4ProxyC),通过基于用户访问情况自适应伸缩缓存的分配方案,使流媒体后缀部分在边缘服务器中缓存的数据段与其流行度成正比。理论分析及实验结果表明,混合流媒体缓存调度策略的实施能有效地降低骨干网络带宽资源消耗,对用户请求到达速率的变化具有良好的适应性。

关键词 流媒体, 补丁预取, 缓存替换, 调度算法

中图分类号 TP393 **文献标识码** A

Dynamic Scheduling Algorithms for Streaming Media Based on Hybrid Content Delivery Network

YE Jian-hong^{1,2} YE Shuang^{1,3}

(School of Computer Science & Technology, Huaqiao University, Xiamen 361021, China)¹

(School of Mechano-electronic Engineering, Xidian University, Xi'an 710126, China)²

(School of Computer Science & Technology, Nanjing University of Science and Technology, Nanjing 210094, China)³

Abstract A hybrid content delivery network combining complementary advantages of CDN and P2P called HyCDN for streaming media was presented. The CVCR4P2P (Comprehensive Value Cache Replacement Algorithm for P2P) algorithm was proposed for the peers inside domain, which considers bytes benefit of prefix data, transmission cost and access rate of streaming media. Another algorithm, DSA4ProxyC (Dynamic Scheduling Algorithm for Proxy Caching), which joints the proxy caching and server scheduling strategies for proxies between domain was also shown. It employs the scheme of cache allocation based on the current batching interval that has non-zero requests, which can be updated periodically according to the popularity of streaming media object. The principle is obeyed that the data cached for each streaming media object are in proportion to their popularity at the proxy server. Theoretical analysis and simulation results show that the hybrid dynamic scheduling can effectively reduce server and network bandwidth usage, and also has a very good adaptability for the variety of the request arrival rate.

Keywords Streaming media, Patch prefetching, Cache replacement, Dynamic scheduling algorithm

1 引言

近年来,Internet上支持视频和音频点播服务(Video on Demand, VoD)的商业产品已有了较快的增长。良好的VoD系统必须具有足够的资源,如CPU处理时间、网络带宽、磁盘存储容量等,以应对用户对高清视频的实时点播要求。随着CPU处理能力的增强,网络带宽及磁盘存储容量逐渐成为VoD系统服务的瓶颈。采取有效的缓存机制不仅能够降低系统服务延时,还能减轻磁盘存储容量的压力,同时增强系统的并发服务能力。

目前对缓存调度算法的研究主要基于两个方面,一是通过在靠近用户的网络边缘部署代理服务器(Proxy),构建一个分布式的流媒体分送服务体系结构,用于缓存并转发流媒体对象,利用最优批处理补丁或前缀缓存^[1-4]方法,与调度算法^[5-8]相结合,在代理服务器中缓存流媒体对象的前缀部分,通过静态或动态的调度方案来降低或消除客户端的启动延时。目前的节目前缀代理缓存所提出的调度方案存在的共同局限是未考虑节目后缀(suffix)的缓存,当客户请求较多时,后缀的传输也会导致很大的骨干网络带宽消耗。

另一种缓存研究定位于客户端,借鉴P2P技术,每一个

到稿日期:2012-05-08 返修日期:2012-08-16 本文受厦门市科技局科技计划创新项目(3502Z20103027),华侨大学科研启动基金项目(09BS514),中央高校基本科研业务费专项资金项目(JB-ZR1130),华侨大学科研基金项目(11HZR05)资助。

叶剑虹(1976-),男,博士后,副教授,CCF会员,主要研究方向为大规模分布式并行计算、故障检测、Petri网理论及应用,E-mail:leafever@163.com;叶双(1977-),女,博士生,讲师,主要研究方向为 workflow 及流媒体网络。

用户节点(peer)既缓存下载数据,又为其他用户上传数据,这种机制增加了整个网络的缓存大小,同时将以前缓存服务器的巨大流量分担到每个 peer 节点。文献[9]提出的 P2P 流媒体结构,使得分层流媒体在 cache 之间的交互达到最优;文献[10]设计了一种 P2Proxy,即只有当用户无法从所属的组中获取数据时,才从服务器获取数据。除了与传统的 VoD 相结合,P2P 的应用还进一步地延伸到实时流媒体领域。文献[11]借助马尔科夫链转移概率矩阵来计算分片在下一时刻的缓存价值以便替换;文献[12]同时考虑了片段最后一次命中时间和片段剩余次数对命中率的影响。现有 P2P 流媒体缓存算法主要集中于考虑缓存代价问题,而对于缓存数据的副本供求关系、数据的紧迫性等对缓存空间的影响及数据命中率关注得并不太多。

本文将融合 CDN 及 P2P 技术各自的优势,提出一种混合式内容分发网络系统结构(Hybrid Content Delivery Network, HyCDN),并在此基础上针对不同区域提出相应的不同的缓存算法,即在域内实现 CVCR4P2P 缓存替换算法,域间实现基于用户请求访问的自适应补丁预取及调度算法 DSA4ProxyC。

本文第 2 节给出混合模式的流媒体系统结构;第 3 节介绍客户端缓存替换算法;第 4 节提出自适应缓存调度算法;第 5 节给出实验仿真;最后总结全文。

2 混合式流媒体体系结构

系统在 CDN 网络中引入 P2P 自治域,采用分区域自治策略的设计思想。用户节点根据其地理区域划分成不同的域,每个域中设置一个边缘服务器。每个域中的边缘服务器以源服务器的身份负责从区域内的 DVS 视频服务器(Digital Video Server)获取数据并转发,同时还负责为区域内的 peer 节点转发来自其他边缘服务器的视频数据。域间网络以 CDN 加入 P2P 模式组织,域内网络以 P2P 模式组织。系统拓扑结构如图 1 所示。

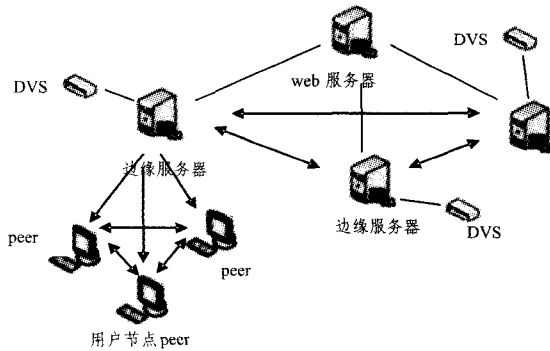


图 1 HyCDN 系统拓扑结构

传统的 CDN 中只有一个源服务器,其余的均为处于边缘的代理服务器。通过 push 或 pull 在源服务器与代理服务器之间传送数据。HyCDN 的域间网络并没有固定的源服务器,所有的边缘服务器既可以充当源服务器,也可以充当代理服务器,角色的转换由一个索引服务器(Web 服务器)决定。边缘服务器可同时从源服务器和其他边缘服务器中获取副本数据。多边缘服务器的方案分担了传统的 CDN 网络中单个源服务器的负载压力。

域内网络中,边缘服务器既作为该域的源服务器,又作为

域中集中式 P2P 的中心索引服务器。peer 节点可以从边缘服务器获取数据,也可以通过 P2P 网络从其他 peer 节点获取数据。

3 P2P 客户端缓存替换算法

域内客户节点需要某媒体数据时,首先向边缘服务器发送请求,经过服务器认证后,服务器把拥有该数据的缓存分片信息的 BM(Buffer Map)列表发送给请求者,请求者按照一定的算法与某些对等节点进行通信,得到相应的媒体数据。该节点获得该媒体数据后,要判断该段数据是否需要缓存。如果需要缓存,则先判断缓存空间是否满,如果不满,则缓存;如果缓存空间已满,则根据相应的缓存算法进行缓存替换,同时修改相应的边缘服务器中的 BM 列表。

常用的缓存替换算法有:LRU(Least Frequently Used),LFU(Least Recently Used),FIFO(First Input First Output)。LRU 只考虑最近请求次数,表面上可以增加热门频道的片段在缓存中的停留时间,但实际上仍存在缓存污染问题,即过去曾经多次使用的文件即使以后不再使用,也仍然占据缓存空间;LRU 只考虑最后的请求时间,并没有考虑频道的热度;FIFO 直观上能够体现出良好的用户实时同步性,但无法区分频道的热门程度。单一的尺度作为是否进行缓存替换的判断标准显然不能很好地满足现在用户多样性的需求,必须设计一种新的能够综合考虑多种因素的缓存替换算法应用于大规模的流媒体系统中。

根据研究和统计表明^[13],节目流行度服从 Zipf-like 分布,即 80% 的用户倾向于访问 20% 的热点文件;而且同一个对象访问中,有相当多的一部分请求只播放了前缀部分。将热门媒体文件的前缀部分尽可能地缓存在 HyCDN 系统的域内客户端,这样既可以减少骨干网的访问压力,又能提高用户的访问时效。热门媒体文件的后缀部分及其它未在客户端缓存的节目将依照第 4 节的规则存储在边缘服务器或源服务器上。

假设媒体流播放速率为 v , t 时刻所有的媒体文件前缀集合为 $PrefixData(t) = \{1, 2, 3, \dots, m\}$,规定每个文件的前缀大小为定长,均为 k ; $\eta_i(t) \in \{0, 1\}$,0 表示 t 时刻前缀文件 i 没有被客户端缓存,1 表示缓存。每个前缀文件 i 的价值函数可以表示为:

$$f_i(t) = S_i \cdot C_i \cdot U_i(t) \quad (1)$$

式中, S_i 表示缓存空间利用率, C_i 为获取数据文件 i 的传输代价, $U_i(t)$ 表示当前文件 i 的热度。缓存空间大小为 C ,最大化缓存空间的利用率实质上是一个最优装载问题^[14],即:

$$\max \sum_{i=1}^m f_i(t) \cdot \eta_i(t) \quad \text{s. t.} \quad \sum_{i=1}^m k \cdot \eta_i(t) \leq C$$

采用贪心算法求解,即价值函数最高的前缀文件先装的策略,可产生最优装载问题的最优解。

对上述引入的缓存空间利用率、传输代价、文件热度概念逐一作详细说明:

$$S_i = \frac{v \cdot Tsum_i}{N_i \cdot k} \quad (2)$$

式中, $Tsum_i$ 是文件 i 总的播放时间, N_i 为访问 i 的总次数。这个参数反映了数据实际被访问状况。

传输代价表明了获取文件 i 的代价:

$$C_i = \alpha C_s + \beta C_p \quad (3)$$

式中, C_s 表示服务器和用户之间传输文件 i 的开销, C_p 表示

文件 i 在客户之间传输所需开销。 α 和 β 的值由 t 时刻文件 i 的需求值 $Req_i(t)$ 及已在用户节点存储的副本数 $Sup_i(t)$ 决定,若 $Sup_i(t)/Req_i(t) \leq 1$, 则 $\alpha=1$; 否则 $\beta=1$ 。理论上讲,传输代价越高,数据被替换的可能性越小。

t 时刻,文件 i 的热度函数为:

$$U_i(t) = \frac{Tsum_i}{t_{i_last} - t_{i_first}} \cdot \min\left\{1, \frac{(t_{i_last} - t_{i_first})/N_i}{t - t_{i_last}}\right\} \quad (4)$$

式中, t_{i_first} 、 t_{i_last} 分别代表文件 i 的第一次访问时间及最后一次访问时间; $\min\left\{1, \frac{(t_{i_last} - t_{i_first})/N_i}{t - t_{i_last}}\right\}$ 表示以后访问的可能性。

上述一些参数的获取需要实际运行时该域的中心索引服务器即负责该域的边缘服务器对数据的访问情况做统计,以便用户节点对自己所存储的前缀文件块做价值判断。

算法 1 综合价值缓存替换算法 (Comprehensive Value Cache Replacement Algorithm for P2P, CVCR4P2P)

输入: t 时刻,某客户端中存储的文件已有 n 个,现有文件 j 需缓存;

输出: 文件 j 是否被缓存;

if ($n \cdot k < C$) 将文件 j 加以缓存;

else {

 客户端中现缓存的文件集不妨标记为 $PrefixData(t) = \{1, 2, 3, \dots, n\}$;

 for ($i=1; i \leq n; i++$) 计算 $f_i(t)$;

 调用快速排序 (Quick sort) 实现 $f_i(t)$ ($i=1, \dots, n$) 排序;

 if ($\min(f_i(t)) > f_j(t)$) 文件 j 不被缓存;

 else 清除 $f_i(t)$ ($i=1, \dots, n$) 中价值最小的段, 缓存 j ;

}

4 边缘服务器补丁预取及调度算法

4.1 域间网络设计

一般而言,代理服务器位于接入网的上层,通常它与客户端处于一个 ISP 域内,所以从代理服务器到客户端可以进行网络层组播。代理服务器间的情况相对复杂,要在其中实现网络层组播比较困难,为此,HyCDN 糅合了 P2P 应用层组播的树型分发技术,构建了域间覆盖网络。每个边缘服务器都接有相应的 DVS,在某个视频源的请求中充当源服务器的边缘服务器,对于另一个视频源的请求来说也许就充当了代理的角色。

域内网络的内容获取如图 2 所示,边缘服务器 X 向拥有某视频源 (DVS_A) 的边缘服务器 A (此时作为源服务器) 发出对视频文件 I 的请求, A 将其信息在索引表中登记,并检索其维护的索引表,返回服务器列表,以表明哪些边缘服务器拥有所请求文件的副本。 X 可以决定从 A 下载,或从其他拥有该视频文件副本的边缘服务器如 Y 中获取数据。

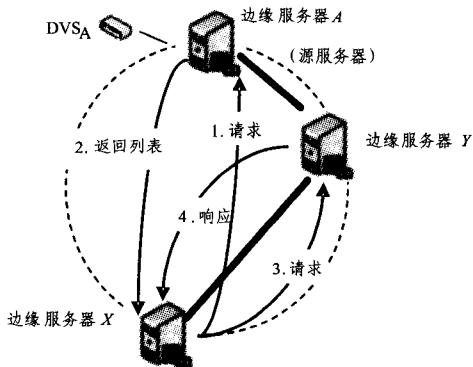


图 2 边缘服务器的内容分发流程

4.2 边缘服务器的补丁预取及调度算法

为进一步减少用户的访问延时,需要将热门视频文件在边缘服务器加以缓存,选择存储哪些流媒体作为副本及如何在边缘服务器进行缓存都是具有挑战性的问题。媒体文件通常非常大,如果将它作为一个整体来对待,要么全部缓存,要么不缓存,则在一个边缘服务器中只能缓存有限的几个媒体对象,这就降低了用户请求的命中率。而且,当有用户正在请求该媒体对象时,在不中断服务的情况下无法进行替换,也降低了缓存空间的利用率,因此在边缘服务器中,需要将媒体对象进行分段缓存。图 3 给出了流媒体对象分段和分块表示。

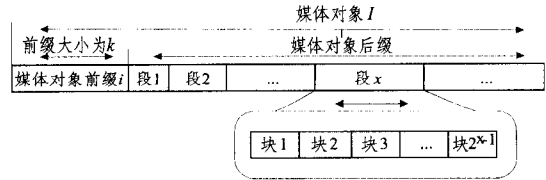


图 3 流媒体对象的分段和分块

用 b 表示每个数据块,其是流媒体文件分割的最小单位;段由若干块组成,它随着文件的向后分割以指数阶增长; $L_x(I)$ 表示文件 I 的后缀数据中第 x 段长度, $L_x(I) = 2^{x-1}b$ 。

通过第 3 节的缓存及替换,一些热点节目的前缀已被缓存在客户端,节目的后缀及非热点节目应采用一定的方案有选择地存贮在边缘服务器中,最终使得每一个媒体对象在边缘服务器中的预存储与其流行度成正比。为实现这一点,本文提出一种基于边缘服务器的自适应缓存分配算法。

算法 2 基于边缘服务器缓存的自适应缓存调度算法 (Dynamic Scheduling Algorithm for Proxy Caching, DSA4-ProxyC)

输入: 在 t_0 时刻有用户对媒体文件 I 发出请求, I 的总长度为 T_I ;

输出: I 在边缘服务器中的缓存情况。

步骤 1 客户端向管辖自己的边缘服务器 X 索取视频文件 I , 转向第 2 步。

步骤 2 若边缘服务器 X 中没有视频文件 I 的信息, 则 X 向拥有该视频文件的边缘服务器申请数据。不妨假定向服务器 A 获取数据: X 请求 A 启动一个组播通道或单播通道向 X 发送视频文件 I 的前缀部分 i , 同时 X 分配一个大小等于 k 的缓存单元, 用于缓存即将到来的前缀文件 i , 转向第 3 步。

步骤 3 X 开通组播通道向客户端发送文件 i , 若在 k/v 时间段内又有新的用户请求到达, 则新用户直接加入文件 i 的组播通道, X 同时为新用户开通一个单播通道传输缺失的前缀部分, 转向第 4 步。

步骤 4 客户端调用 CVCR4P2P 算法判断自己是否缓存文件 i , 若有客户端缓存了文件 i , 则将 X 中原用于存储 i 的空间释放, 否则保留 X 中对 i 的缓存, 转向第 5 步;

步骤 5 若申请前缀 i 的用户并未完成对整个前缀部分的播放, 转向第 9 步; 否则, 在 $t_0 + (k-b)/v$ 时刻, 客户端向 X 请求文件 I 的后缀部分, 转向第 6 步。

步骤 6 X 请求 A 启动一个组播通道或单播通道向 X 发送视频文件 I 的后缀部分, 同时 X 分配一个大小等于 b 的缓存单元, 用于缓存即将到来的后缀文件段 $L_1(I)$, 转向第 7 步。

步骤 7 对剩余 I 的后缀部分的存储依照以下原则: 若在每个后续时间间隔 $(t_y \sim t_y + L_y(I)/v)$ 内, $1 \leq y \leq \varphi$, φ 的取值参见式 (6), X 总会收到新客户对 $L_y(I)$ 的请求, 则 X 自动延续分配一个 $2^y b$ 的空间用于存储随后的 $L_{y+1}(I)$ 文件。如此反复, 直至① X 缓存了整个文件 I 的所有后缀; ② X 在某个时间间隔 $(t_y \sim t_y + L_y(I)/v)$ 内没有收到新客户的请求, $L_{y+1}(I)$ 至

文件的结尾部分都不再被 X 缓存,转向第 8 步。
 以下给出 φ 的取值范围, $L_\delta(I)$ 为 I 的最后一段,有:

$$T_I = \begin{cases} k + \sum_{r=1}^{\varphi} L_r(I) + L_\delta(I), & L_\delta(I) < L_\varphi(I) \\ k + \sum_{r=1}^{\varphi} L_r(I), & L_\delta(I) = L_\varphi(I) \end{cases} \quad (5)$$

已知 $L_r(I) = 2^{r-1}b$, $\sum_{i=0}^{\varphi-1} 2^i = 2^\varphi - 1$, 求得 φ 的近似解为:

$$1 \leq \varphi \leq \ln \frac{T_I - k}{b} \quad (6)$$

步骤 8 X 将接收到的后缀文件通过组播通道向客户端转发,直至完成对客户的播放请求,转向第 9 步。

步骤 9 算法结束。

图 4 给出了文件 I 在边缘服务器的缓存及组播方式。由于用户请求播放时间长于文件 I 的前缀部分,因此 X 在 $(t_0 + (k-b)/v \sim t_1)$ 时刻会预存 $L_1(I)$ 数据段; $(t_1 \sim t_2)$ 、 $(t_2 \sim t_4)$ 时刻 X 都收到了新用户发出的分别对 $L_1(I)$ 、 $L_2(I)$ 的请求,所以 $L_2(I)$ 、 $L_3(I)$ 也被 X 缓存; $(t_4 \sim t_8)$ 期间 X 没有收到任何关于 $L_3(I)$ 的新请求, I 中其后的数据段将不再被 X 缓存; $(t_8 \sim t_0 + T_I/v)$ 期间的流媒体文件都将直接通过组播通道转发给用户,即便 X 在 $(t_l \sim t_{l+1})$ 时刻还接收到了新用户的请求,无论该请求是对 I 后缀中的哪一段进行访问请求, X 都不做缓存处理。图 4 显示 X 最终存储了 I 后缀中共长为 $7b$ 的部分。

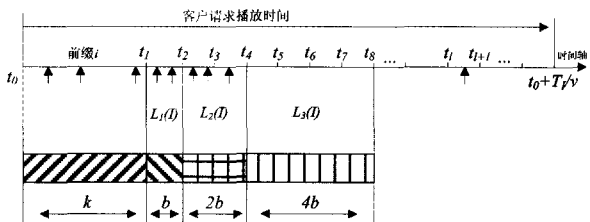


图 4 DSA4Proxy 算法服务示意图

5 实验仿真

设域间流媒体系统共有 10 个边缘服务器,每个边缘服务器拥有 100 个 CBR(Constant Bit Rate)流媒体对象,媒体对象的播放速率为 1.5Mbps(H. 264 编码)。为方便比较实验结果,令每个媒体对象等长,播放持续时间均为 90min,流媒体对象的访问概率服从参数 $\theta=0.271$ 的 zipf-like 分布^[15],前缀大小 $k=90\text{M}$,后缀数据块 b 为 30M。通过对商业流媒体视频网站¹⁾连续一周用户的加入和离开数据的监测发现,24 小时内用户节点加入域内的速率 $\lambda(t)$ 均值在 2~107 人/s 波动,离开网络速率 $\beta(t)$ 值约为 1.3~120 人/s,其中,异常离开率为 0.05 人/s,满足非齐次泊松(Poisson)分布。采用 NS2 作为仿真工具,运行平台为 Windows XP 系统,CPU2.7GHz,内存 2GB,对本文的 CVCR4P2P+DSA4ProxyC 混合缓存调度算法与文献[16]的 P³S²A 算法在播放启动延时和服务器负载方面进行了对比实验。

图 5 显示的是用户节点的平均启动时延。从图 5 可以看出,随着网络规模的增大,两种算法的节点平均启动时延都迅速增大,然后逐步减低并趋于稳定的趋势。这源于两种算法都采用将热门节目的前缀部分前推缓存的思想,不同之处在于 CVCR4P2P+DSA4ProxyC 混合缓存调度算法将节目前缀推送至客户端进行缓存,P³S²A 算法将前缀缓存在代理服务

器中。初始网络规模不大时,两种算法都直接从源服务器获取数据,随着网络规模的扩大,从邻近节点以 P2P 的方式获取节目前缀的效率显然高于从代理服务器获取数据,这一点在图 5 中得到验证,特别地,当网络中节点个数超过 1000 时,受限于代理服务器处理能力,P³S²A 算法的时延又有所升高。

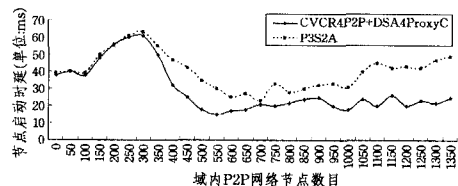


图 5 节点平均启动时延

图 6 的横坐标同样表示用户节点数,纵坐标表示服务器负载(服务器输出数据库的数量/系统吞吐量 * 100%)。图 6 显示两种算法的服务器负载都会随着网络规模的增加而增长,并最终趋于稳定。混合缓存调度算法比 P³S²A 算法更快进入稳定,服务器负载也相对较低,这得益于混合缓存调度算法是根据每一个批处理区间的访问请求到达情况来决定是否进行下一阶段的补丁预取缓存,从而最终实现服务器中只缓存流行节目的部分。同时,节目前缀的客户端缓存也避免了网络规模增大时服务器的工作压力,进一步减少了数量交换量。

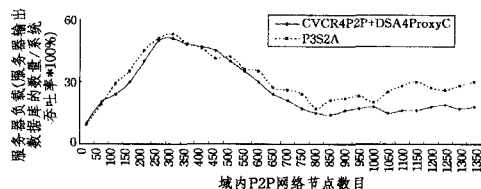


图 6 服务器负载

结束语 鉴于 CDN 和 P2P 网络的互补优势,本文提出一种混合内容分发网络 HyCDN,并在此基础上给出混合缓存调度算法。与传统算法相比,在不消耗更多的缓存空间情况下,本文提出的算法能够更有效地提供用户访问体验,并节省骨干网络的带宽资源,特别在访问请求强度较高的情况下,效果更为突出。当系统规模扩大以后,多个边缘服务器之间的缓存协作也是提高系统效率的有效手段之一,这将是未来课题研究的方向之一。

参考文献

- [1] Sen S, Rexford J, Towsley D. Proxy prefix caching for multimedia streams[C]//IEEE INFOCOM. New York: IEEE Computer Society Press, 1999: 1310-1319
- [2] Guo Y, Subhabrata S, Towsley D. Prefix Caching assisted Periodic Broadcast: framework and Techniques to Support Streaming for Popular Videos[R]. Technical Report TRO1-22, IEEE ICC, 2001
- [3] Bing W, Subhabrata S, Adler M, et al. Proxy-based Distribution of Streaming Video over Unicast Multicast Connections[R]. New York: University of Massachusetts Amherst, 2001
- [4] Eager DL, Ferris MC, Vernon MK. Optimized Caching in Systems with Heterogeneous Client Populations [J]. Performance Evaluation, Special Issue on Internet Performance Modeling, 2000, 42(9): 163-185

(下转第 83 页)

¹⁾ <http://www.cn-topnet.com/>

FECA 规则的阈值自适应学习机制是进一步的研究工作。本文的后续工作主要是对普适访问控制的 FECA 规则的执行模型进行探讨,并分析和研究 FECA 规则集的静态结构特性和动态行为特性(如一致性、可终止性)以及 FECA 规则的阈值自适应学习机制。

参 考 文 献

- [1] Weiser M. The computer for the twenty-first century[J]. *Scientific American*, 1991, 265(3): 94-104
- [2] Weiser M. Hot topics-ubiquitous computing[J]. *IEEE Computer*, 1993, 26(10): 71-72
- [3] 徐光祐, 史元春, 谢伟凯. 普适计算[J]. *计算机学报*, 2003, 26(9): 1042-1050
- [4] Hilary H. Security is fuzzy; applying the fuzzy logic paradigm to the multi-policy paradigm[C]//*Proceedings of the ACM Workshop on New security Paradigms*. New York: ACM, 1993: 175-184
- [5] Chang E, Thomson P, Dillon T, et al. The fuzzy and dynamic nature of trust[J]. *Lecture Notes in Computer Science*, 2005, 3592: 161-174
- [6] 李小勇, 桂小林. 大规模分布式环境下动态信任模型研究[J]. *软件学报*, 2007, 18(6): 1510-1521
- [7] Ishikawa H, Kubota K. An active object-oriented database: a multi-paradigm approach to constraint management[C]//*proceedings of 19th VLDB*, 1993, 1993: 467-478
- [8] Gehani N, Jagadish H, Smueli O. Event specification in an active object-oriented database[C]//*Proceedings of the ACM-SIGMOD International Conference on Management of Data*, 1992. CA: ACM, 1992: 81-90
- [9] 李凤华, 苏铨, 史国振, 等. 访问控制模型研究进展及发展趋势[J]. *电子学报*, 2012, 40(4): 805-813
- [10] 魏志强, 康密军, 贾东宁, 等. 普适计算隐私保护策略研究[J]. *计算机学报*, 2010, 33(1): 128-138
- [11] 林莉, 怀进鹏, 李先贤. 基于属性的访问控制策略合成代数[J]. *软件学报*, 2009, 20(2): 403-414
- [12] 盖新貌, 沈昌祥, 刘毅, 等. 基于属性访问控制的 CSP 模型[J]. *小型微型计算机系统*, 2011, 32(11): 2217-2222
- [13] 王小明, 付红, 张立臣. 基于属性的访问控制研究进展[J]. *电子学报*, 2010, 38(7): 1660-1667
- [14] 崔永泉, 洪帆, 龙涛, 等. 基于使用控制和上下文的动态网格访问控制模型研究[J]. *计算机科学*, 2008, 35(2): 37-41
- [15] 蒲芳, 姜涛, 曹奇英. 普适计算的上下文访问控制模型[J]. *计算机应用研究*, 2009, 26(1): 317-320
- [16] 窦文阳, 王小明, 张立臣. 普适环境下的动态模糊访问控制模型研究[J]. *计算机科学*, 2010, 37(9): 63-67
- [17] 王小明. 面向普适计算的区间值模糊访问控制[J]. *计算机科学与探索*, 2010, 4(10): 865-880
- [18] 邓勇, 张琳, 王汝传, 等. 网格计算中基于信任度的动态角色访问控制的研究[J]. *计算机科学*, 2010, 37(1): 51-54
- [19] 徐文拴, 辛运伟, 卢桂章, 等. 普适计算环境下信任管理模型的研究[J]. *计算机科学*, 2009, 36(2): 103-106, 133
- [20] 董理君, 余胜生, 杜敏, 等. 一种基于环境安全的角色访问控制模型研究[J]. *计算机科学*, 2009, 36(1): 51-54, 59
- [21] 戴刚. 基于使用控制和上下文的模糊访问控制模型研究[D]. 重庆: 重庆大学, 2009
- [22] 翟浩良, 韩道军, 李磊. 基于情景演算的动态访问控制模型[J]. *计算机科学*, 2012, 39(6): 35-39
- [23] 吴新松, 贺也平, 周洲仪, 等. 一个环境适应的基于角色的访问控制模型[J]. *计算机研究与发展*, 2011, 48(6): 983-990
- [24] 贺春林, 滕云, 彭仁明. 一种基于 ECA 规则的 Web Service 工作流模型的研究[J]. *计算机科学*, 2009, 36(8): 112-115
- [25] 熊伟, 吴焯, 张震, 等. 基于触发路径的主动规则集终止性分析[J]. *计算机学报*, 2012, 35(1): 65-75
- [26] 曾文艺, 于福生, 李洪兴. 区间值模糊推理[J]. *模糊系统与数学*, 2007, 21(1): 68-74
- [27] 王国俊. 三 I 方法与区间值模糊推理[J]. *中国科学(E 辑)*, 2000, 30(4): 331-340

(上接第 64 页)

- [5] Dan A, Shahabuddin P, Sitaram D. Scheduling policies for an on-demand video server with batching[C]//*Proc of ACM Multimedia*. New York: ACM press, 1994: 168-179
- [6] Hua K, Cai Y, Sheu S. Patching: A multicast technique for true video-on-demand services [C]//*Proc of ACM Multimedia*. New York: ACM press, 1998: 12-16
- [7] Carter S W, Long D E. Stream Tapping: a System for Improving Efficiency on a Video-on-Demand Server [R]. UCSC-CRL-97-11. Univ. of California, Santa Cruz: November 1997
- [8] Viswanathan S, Imielinske T. Metropolitan area video-on-demand service using Pyramid Broadcasting [J]. *IEEE Multimedia Systems*, 1996, 51(4): 197-208
- [9] Ernst-Desmulier J B, Charlet D, Chatonnay P, et al. A Peer to Peer approach for cache sibling[C]//*First International Conference on Distributed Frameworks for Multimedia Applications*. Besancon, France, 2005: 323-330
- [10] Lee G J, Cho i C K, Chi C Y, et al. P2Proxy: Peer to Peer proxy caching scheme for VOD service[C]//*Sixth International Conference on Computational Intelligence and Multimedia Applications*. 2005: 272-277
- [11] 杨静, 李润知, 王宗敏. 基于时间间隔的 P2P 流媒体直播系统缓存算法[J]. *计算机工程与设计*, 2010, 31(1): 90-93
- [12] 胡懋智, 徐恪, 夏树涛, 等. TOW: 一种新的 P2P 实时流媒体缓存替换算法[J]. *小型微型计算机系统*, 2009, 30(8): 1484-1489
- [13] Cherkasova L, Gupta M. Analysis of enterprise media server workloads: access patterns, locality, dynamics, and rate of change [EB/OL]. <http://www.hpl.hp.com/techreports/2002/HPL-2002-56.html>, 2002-03-26
- [14] 王晓东. 算法设计与分析[M]. 北京: 清华大学出版社, 2003: 114-116
- [15] Wang B, Sen S, Adler M, et al. Optimal proxy cache allocation for efficient streaming media distribution [J]. *IEEE Transactions on Multimedia*, 2004, 6(2): 366-374
- [16] 覃少华, 李子木, 蔡青松, 等. 基于代理缓存的流媒体动态调度算法研究[J]. *计算机学报*, 2005, 28(2): 185-194