

部分缓存算法在流媒体内容分发网络中的应用研究

冯欣¹ 王康² 邓一贵¹

(重庆大学计算机学院 重庆 400044)¹ (重庆大学网络中心 重庆 400044)²

摘要 内容分发网络是互联网中解决响应速度、提高服务质量的一种应用技术,而流媒体内容分发网络是将内容分发网络应用于流媒体传输领域。流媒体内容分发网络中的边缘服务器缓存策略是流媒体内容分发网络的核心。本文深入研究了现有各种缓存策略;在流媒体内容分发网络中,提出了一种部分缓存算法。实践证明,这种部分缓存策略能明显提高流媒体内容分发网络的并发性能、命中率和服务质量等性能。

关键词 流媒体内容分发网络,边缘服务器,部分缓存,命中率,服务质量

Algorithm of Part-Cache Strategy and its Application in Streaming Media Content Delivery Network

FENG Xin¹ WANG Kang² DENG Yi-Gui¹

(Department of Computer, Chongqing University, Chongqing 400044)¹ (Network Center, Chongqing University, Chongqing 400044)²

Abstract Content delivery network is an internet application aiming to solve the response rate and improve the quality of service in recent years. Streaming media content deliver network is the application of streaming media technology in content delivery network. The cache strategy is the core algorithm in edge server proxy. So based on deep research on the current cache algorithms, this paper presents a new partial cache algorithm. This algorithm can obviously improve the system Concurrency, hit rate and QoS in practical project.

Keywords Streaming media content delivery network, Edge server proxy, Partial cache hit rate QoS

1 引言

1.1 流媒体内容分发网络

Internet 的迅速发展使得信息通信发生了革命性的变化,也使各种网络技术不断走向成熟。内容分发网络(Content Delivery Network,简称 CDN)正是在为解决目前的网络拥塞、带宽瓶颈等问题而生,并在短时间内得到迅速发展的一种特殊的网络体系结构。CDN 不仅可以为静态的 Web 内容提供快速高效的服务,而且更多地体现在其动态内容和视频、音频等流媒体(Streaming Media)应用上。

流媒体内容分发网络(简称 SMCDN)就是在流媒体数据承载网上建立内容分发网络,将 CDN 技术应用于流媒体传输领域。CDN 能够让流媒体内容迅速推到网络边缘,使用户的请求能够就近得到快速响应,这不仅大大减轻了原服务器和网络的负载,而且降低了网络拥塞出现的概率,缩短了请求时延。SMCDN 在继承普通流媒体技术“边接收边播放”模式的基础上,提供了更优质快速的服务,不仅解决了普通流式技术“先发后至”、丢包、原服务器瓶颈等问题,而且融合了普通流式技术减少客户端资源、保护版权等优点,有较为广泛的应用前景。

1.2 流媒体内容在 CDN 边缘服务器缓存的现状研究

在 SMCDN 中,原服务器的流媒体内容被分发到接近用户的网络边缘,由边缘服务器(Edge Server Proxy,简称 ESP)进行缓存。媒体文件在 ESP 中的缓存策略是提高流媒体内容分发系统性能的关键。从目前动态内容在代理服务器的缓存方面的研究成果来看,按照副本粒度(Replication Granularity)^[4]粗细可大致分为整体备份(Entire replication)和对象备份(Object replication)^[4]。

在整体备份策略中,原服务器将自己的全部对象集都复制给代理服务器,副本粒度是最粗的^[6,7]。整体备份如果只面向静态内容存取,性能是可以满足的,但对于容量很大的动态内容,存在以下问题:a)对代理服务器存储容量要求很高;b)时间和带宽开销很大;c)有些内容是用户很少访问甚至一直不会访问的,造成了代理服务器空间浪费。

与整体备份相反,对象备份是以单个媒体文件为对象进行缓存,粒度比较细^[8]。目前的 SMCDN 系统就是采用面向对象备份的缓存策略,很好地解决了上述的第三个问题。然而,对象备份虽然在一定程度上缓解了整体备份的问题,但对于单个文件对象仍然是将一个文件作为整体缓存,如文[1~4]中流媒体内容在 ESP 上的缓存,技术上都是将整个媒体文件作为一个整体来缓存,因此仍然存在上述的前两个问题,并没有很大地改善 SMCDN 系统性能。

目前在流媒体技术领域已经出现了将一个媒体文件整体划为部分进行缓存的策略,如文[5,9]中的前缀后缀法、文[5]中的分段缓存法,但这些方法都没有应用在 CDN 系统中,没有将流媒体技术特点和 CDN 的优势相结合。本文在深入研究了以上策略的基础上,提出一种部分缓存策略,在 ESP 中缓存一个文件的部分内容,能使副本粒度更细,同时保证在缓存相同节目的情况下,可以缓存更多的文件,服务质量也不会降低。实践证明,这种策略能有效节省空间,减小带宽,提高整个 SMCDN 系统的性能。

2 SMCDN 系统的工作原理

CDN 是通过在现有网络体系中增加一层特殊的网络架构,将网站的内容推到最接近用户的网络“边缘”,用户就可以在“最近”的位置快速访问到所需的内容。用户在从发出请求

到获取到访问内容的过程中不需要任何参与, CDN 负责了全部工作。因此, 与需要用户自己选择镜像点的镜像技术相比, CDN 提供了更加透明且更快速的服务。

SMCDN 系统是将 CDN 专门用在互联网上快速高效地传输流媒体信息, 它主要由三大模块组成: 边缘服务器(Edge Server Proxy)、内容路由服务器(MediaDNS)和内容管理服务器(CManager)(图 1)。

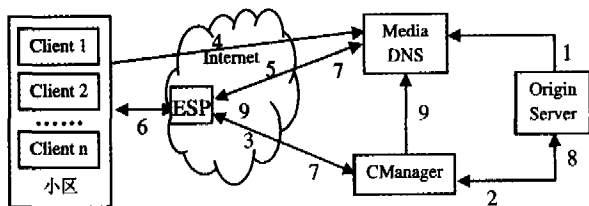


图 1 SMCDN 系统架构

如图 1 所示, SMCDN 系统的工作流程如下: (1) 原服务器将自己的 URL 向内容路由系统(MediaDNS)注册; (2) 原服务器同时向内容管理系统(CManager)发布内容; (3) CManager 将内容分发到网络边缘的 ESP 上; (4) 小区用户提出内容请求; (5) 用户的请求经由 MediaDNS 重定向到离用户最近的 ESP; (6) 由该 ESP 为用户提供服务; (7) ESP 定期向 MediaDNS, CManager 汇报缓存和负载情况; (8) CManager 将各个节点(ESP)的管理情况反馈给原服务器中心数据库; (9) 给各个 ESP 及 MediaDNS 提供节目信息。

ESP 服务器的核心功能是缓存原始服务器的内容, 给用户提供服务, 因此内容在服务器上的缓存策略是 ESP 的核心算法。本文第 3 部分将详细分析本文提出的一种新的部分缓存策略算法。

3 边缘服务器的部分缓存算法

3.1 问题描述

在 SMCDN 系统中, 假设 CManager 预先将 m 个节目(记为 $P_f = \{p_1, p_2, \dots, p_m\}$)“推”到 k 个选定的 ESP(记为 $E = \{e_1, e_2, \dots, e_k\}$)上。当用户 U_1 请求访问节目 p_u , MediaDNS 将请求重定向到离 U_1 最近的 ESP e^* (假设只考虑 $e^* \in E$)。如果 e^* 上存在 p_u 的缓存, 即 $p_u \in P_f$, 则请求命中, 此时 e^* 直接为 U_1 提供“就近”服务。而当 $p_u \notin P_f$, 请求未命中, e^* 会先将 p_u 从原服务器“拉”下在本地缓存, 然后再为用户提供服务。

在目前 SMCDN 所普遍采用的对象缓存策略中, 无论请求是否在 e^* 命中, 上述“推”、“拉”过程中的 e^* 上都是缓存节目 p_u 的整个文件 S_u (S_u 表示 p_u 的文件大小, 单位 kB)。

部分缓存策略正是打破这种缓存整个文件的观念, 在保证服务质量不降低的前提下, 只在 ESP 上缓存每个节目的前部分内容。这样, 在“推”过程中, ESP 可以预先缓存更多的节目。假设采用部分缓存法缓存 n 个节目(记为 $P_{part} = \{p_1, p_2, \dots, p_n\}$), 则 $n > m$, 从而用户请求命中率提高。仍然假设用户 U_1 请求访问 p_u , e^* 是离 U_1 最近的 ESP, 采用部分缓存法, e^* 预先缓存 P_{part} 集合中每个元素的前部分内容:

1) 当 $p_u \in P_{part}$ 请求命中, e^* 在用 S_{p1} 提供 p_u 播放服务的同时, 从原服务器取下 p_u 剩下的内容 S_{p2} 缓存(图 2), 在这种“边服务边下载”的模式下为用户提供连续的播放服务。

2) 若 $p_u \notin P_{part}$ 请求未命中, 用户 U_1 的请求被 MediaDNS 重定向到原服务器, 同时启动 e^* 从原服务器“拉”下计算好的

S_{p1} 部分进行缓存, 也就是用户 U_1 的请求此时由原服务器响应, 原服务器为 U_1 提供普通的流媒体服务播放 p_u 。

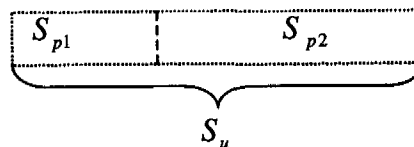


图 2 将 p_u 整个文件分为两个部分

部分缓存算法所要解决的主要问题是如何划分整个文件的两个部分, 才能在保证媒体流连续播放的前提下, 使 ESP 上缓存文件内容最小。此算法问题描述如下:

- 假设 ESP 到用户 U 的带宽为 b_c (单位 bps);
- 当前播放文件的码流所需要的带宽 b_n (单位 bps);
- ESP 到原服务器的带宽 b_s (单位 bps);
- 限制条件: $b_c > b_n$;
- 请求的节目在 ESP 上部分缓存 S_{p1} , 并在 SMCDN 系统上实现 ESP 上的部分缓存。

3.2 算法的基本思路

算法输入: b_c, b_n, b_s, S_u

算法输出: S_{p1}

算法描述:

1) 用户 U_1 提出访问节目 p_u 的请求。MediaDNS 接收到用户的请求后, 根据用户的信息, 把用户的请求重定向到离用户“较近”的 ESP e^* (e^* 上预存了 n 个节目的前部分内容(记为 $P_{part} = \{p_1, p_2, \dots, p_n\}$))。

2) e^* 检查请求是否命中。如果 $p_u \notin P_{part}$ 请求未命中, 则 ESP 把用户 U_1 请求重定向到原服务器, 由原服务器为 U_1 提供服务, 同时转入 3。如果当 $p_u \in P_{part}$ 请求命中, 转入 5。

3) ESP 获取参数: 流媒体文件类型、 S_u, b_n, b_s, b_c 。根据这些参数, 按照以下的算法计算出应该从原服务器“拉”下的文件大小 S_{p1} :

$$a) S_{p1} = S_u - S_{p2} \quad ①$$

b) $S_{p2} = \text{ESP 从原服务器取部分 } S_{p2} \text{ 时的传输速率} \times \text{传输时间} \quad ②$

其中传输速率即为 ESP 到原始服务器的带宽 b_s

c) 由于当 ESP 一开始用 S_{p1} 为用户服务, 即节目一开始播放, ESP 就同时开始从原服务器取下 S_{p2} , 直到整个节目播放完毕。 S_{p2} 刚好取完, ESP 的缓冲区为空, 所以有传输时间 = 节目播放时间 $③$

d) 节目播放时间 = $S_u \times 2^{13} / b_n$ (其中 2^{13} 表示 kB 和 bit 之间的换算关系) $④$

$$e) \text{ 将 } ④ \text{ 代入 } ③, \text{ 得: 传输时间} = S_u \times 2^{13} / b_n \quad ⑤$$

$$f) \text{ 将 } ⑤ \text{ 代入 } ②, \text{ 得: } S_{p2} = b_s \times S_u / b_n \quad ⑥$$

g) 将 $⑥$ 代入 $①$, 加之限制条件, 得:

If ($b_c > b_n$)

$$S_{p1} = S_u - b_s \times S_u / b_n \quad ⑧$$

4) ESP“拉”下所需文件大小 S_{p1} , 保存在磁盘; 转到 6。

5) ESP 把缓存的 S_{p1} 读入缓冲, 在给用户提供服务的同时启动“拉”线程, 从原服务器把 S_{p2} “拉”到该 ESP 的同时把“拉”下的内容连接到缓冲区, 如图 3。



图 3 S_u 文件缓存情况

- S_2 表示已缓存到 ESP 上的内容
- S_1 表示正在从原服务器下载的内容
- S_3 表示还未下载的内容

$$S_u = S_1 + S_2 + S_3$$

6) 流调度完毕, 算法结束。

综上所述, 此算法的关键在于: 缓存的文件大小刚好能够满足连续码流连续播放的要求。当边缘服务器给用户流调度完毕时, 缓存为空。在用户 U_1 请求未命中时, 算法采用让原服务器为用户 U_1 提供普通流媒体在线播放的方式, 也就是用户 U_1 在观看 p_u 时并没有享受到 SMCDN 的好处, 但这样却同时满足了: a) 让 U_1 在其延时容忍时间内就得到服务; b) 触发了 ESP 启动原服务器“拉”下 S_{p_i} 的线程; c) 牺牲一用户的 QoS, 但其他用户 U_2, U_3, \dots 再访问 p_u 时就可以享受到 SMCDN 带来的高质量服务。而在缓存整个文件中, 用户必须等待 ESP 将整个文件下载之后才能得到服务, 这样的延时是用户不能容忍的。

4 实验结果分析

本文针对 SMCDN 提出了基于缓存部分流媒体内容的缓存算法思想, 通过参与“下一代互联网内容分发网络”研究项目, 在基于 IPv6 的网络环境下, 将这种策略在实际系统环境中进行了应用研究, 对其性能进行了测试并与其它的主流 CDN 系统进行了比较(如表 1)。

表 1 SMCDN 系统与其它系统性能比较

性能参数	采用部分缓存的 SMCDN	其它 CDN 系统的最优性能
从用户提出请求到得到反应的时间(单位:s)	<3s	<4s
客户端播放器的缓冲时间(单位:s)	<10s	<10s
播放中延迟(单位:s)	<2s	<5s
支持的网络协议	Ipv4/Ipv6	Ipv4
同时并发提供不同节目的流(单位:个/ESP)	200~300	150
内容路由并发性能(Media DNS 在每秒处理的客户的个数)	230~250	150~200
ESP 的命中率	90%	30%

从表 1 可以得知, 采用部分缓存的 SMCDN 系统, 在同时提供不同节目的流和 ESP 的命中率这两项性能指标方面提高得比较明显, 从而在相同的硬件条件下可以存取更多的流媒体节目。但是, 应用部分缓存策略的 SMCDN 系统, 实现较为复杂, 同时受实际网络情况的影响比较大, 与在良好的实验环境下相比, 在稳定性上有一定的下降。这方面, 我们正在做进一步的研究。

结论 随着用户对网络流媒体需求的不断增加, 人们迫切希望在互联网上看到电视或录像机一样效果的内容, 所以流媒体内容分发网络是达到此目的的选择方法之一。本文提出的部分缓存策略应用到流媒体内容分发网络, 体现了流式技术与 CDN 相结合的优势, 有效提高了系统的命中率和服务质量。由于 CDN 涉及内容广泛, 本文仅仅从 SMCDN 中的 ESP 的缓存策略进行了初步研究, 因此需要不断完善和提高。

参考文献

- 1 胡海清, 傅鹤岗, 朱庆生. 基于软件 Agent 技术的内容分发网络研究[J]. 计算机应用, 2004, 24(6): 51~53
- 2 Lu Jian. Reactive and Proactive Approaches to Media Streaming [C]. In: Information Technology: Coding and Computing, 2001. Proc., Intl. Conf. on 2001. 5~9
- 3 Yoshimur T, Yonemoto Y, Ohya T, et al. Mobile Streaming Media CDN Enabled by Dynamic SMIL [J]. ACM, 2002. 651~661
- 4 Fujita N, Enomoto N, Iwata A, et al. Coarse-Grain Dynamic Replication Schemes for Scalable Content Delivery Networks [J]. IEEE, 2002. 2235~2239
- 5 王薇薇, 李子木. 基于 CDN 的流媒体分发技术研究综述[J]. 计算机工程与应用, 2004(8): 121~125
- 6 Cronin E, Jamin S, Cheng Jin, et al. Constrained Mirror Placement on the Internet [J]. IEEE, 2002, 20(7): 1369~1382
- 7 Qiu Lili, Padmanabhan V N, Voelker G M. On the placement of Web Server Replicas [J]. IEEE, 2001, 3(22~26): 1587~1596
- 8 Kangasharju J. Object replication strategies in content distribution networks [J]. Computer Communications, 2002. 25: 376~383
- 9 Sen S, Rexford J, Towsley D. Proxy Prefix Caching for Multimedia Streams [J]. IEEE, 1999, 3(21~25): 1310~1319

(上接第 138 页)

- 7 Srivastava J, et al. Web usage mining: Discovery and application of usage patterns from Web data. SIGKDD Explorations, 2000, 1(2)
- 8 Zaiane OR, Xin M, Han J. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In: Proc. of Advances in Digital Libraries Conf. (Adl'98), Santa Barbara, CA, 1998. 19~29
- 9 Lieberman H. Letizia: An Agent that assists Web browsing. In: Proc. of the 14th Intl. Joint Conf. on Artificial Intelligence, Montreal, IJCAI'95, Aug. 1995. 924~929
- 10 Balabanovic M, Shoham Y. An adaptive Agent for automated Web browsing. Journal of Visual Communication and Image Representation, 1995, 6(4)
- 11 Mladenic D. Personal Web Watch: Design and Implementation: [Technical Report IJS-DP-7472]. Department for Intelligent System, J. Stefan Institute
- 12 Joachims T, Freitag D, Mitchell T. WebWatcher: a tour guide for the World Wide Web. In: Georgeff M P, Pollack E M, eds.

Proc. of the Intl. Joint Conf. on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1997. 770~777

- 13 http://profusion.itc.ukans.edu
- 14 王耀南. 智能控制系统——模糊逻辑. 专家系统. 神经网络控制, 湖南大学出版社, 1996
- 15 Pandya A S, Macy R B. 神经网络模式识别及其实现. 北京: 电子工业出版社, 1999
- 16 李勇, 桑艳艳. 网络文本数据分类技术与实现算法. 情报学报, 2002, 21(1): 21~26
- 17 Herlocker J L, et al. An Algorithmic Framework for Performing Collaborative Filtering. SIGIR, 1999. 230~237
- 18 Zadeh L A. Fuzzy sets. Information and Control. 1965, 8: 338~353
- 19 Lee K F, Mahajan S. Automatic Speech Recognition: The Development of The SPHINX System. Dordrecht, Netherlands: Kluwer, 1989
- 20 苏中, 马少平, 等. 基于 Web-Log Mining 的 N 元预测模型. 软件学报, 2002, 13(1): 136~141