Web Cache 系统的设计与分布技术

Research on Web Caching Techniques

顾铁成 杨书慧 陆桑璐 陈道蕃

(南京大学计算机软件新技术国家重点实验室 计算机科学与技术系 南京 210093)

Abstract This paper discusses the role of Web caching, analyses some major topics in Web caching research, such as Web cache's architectures, design techniques, deployment policies and inter-cache communication protocols for distributed cache servers, and multimedia proxy caching systems. Besides, some open problems in this field are also presented.

Keywords Internet, Web caching, Proxy server

1 引言

随着因特网技术和应用向广度和深度迅猛发展,人们除了在网共享信息外,还试图通过无所不在的因特网来大规模地共享计算能力和服务等一切可以共享的资源。但是,由于网络中瓶颈的存在及热点资源的相对集中,这种迅猛发展也带来了一个问题,即人们上网时常常体验到的“World Wide Wait”问题。要解决这一问题,一种有效的解决办法就是采用 Web Caching 技术,即将 Web 上常被人们访问的、热门的信息缓存在 Web 服务器与最终用户之间的缓存服务器系统中。

当前,Web Caching 是一个正在迅速发展的领域,无论科研界还是工业界,都在这一方面投入了大量的精力。如果设计和部署得当的话,Web Caching 可以带来很多好处,如节约大量主干网带宽,降低和平衡 Web 服务器的负载,改善 Web 服务器的可获得性,降低用户感知的网络延迟,带来更好的内容可获得性等。事实证明,Web Caching 是有效的,因为目前 Web 上的信息多数是只读的。因此,Web Caching 已成为 Web 的一个重要组成部分,甚至还催生了一种新的产业,即“内容分布与传输网络”(content delivery networks)。

在本文中,我们综合地研究了当前最新的 Web Caching 设计和实现方面的各种系统方案、技术与问题。介绍几种主要的 Web Caching 种类与分布策略;总结了几种 Web Caching 体系结构;介绍了 Web Caching 间的通信机制;讨论了 Web Cache 置换算法;最后对全文进行了小结。

2 Web Cache 系统的分布策略

Web Cache 系统可以分布在 Web 上的不同位置,如 Web 服务器端或客户方的浏览器一端,也可以位于两者中间的某一处。当 Web Cache 位于 Web 上的不同位置上时,其作用也会有所不同和侧重。例如,当缓存位于 Web 服务器一端时,其作用主要是平衡 Web 服务器的负载,增强服务器的可扩展性和可获得性。当缓存接近客户端时,其主要作用是节约网络带宽、改善用户感知的网络性能等。

Web Caching 系统的分布方式一般有三种:1)靠近客户(content consumer, consumer-oriented);2)靠近内容提供商

(content provider, provider-oriented);3)根据用户的访问模式和网络的拓扑、状况等,来选择分布于网络中的一些关键的位置上。下面,我们就来分别讨论这几种分布策略的优劣。

第一种缓存分布策略如在代理缓存(包括透明代理缓存)技术中那样,其长处在于可以对一个用户群利用一个或多个缓存。如果那些用户常常会访问比较相同的内容的话,那么这种分布策略就可以通过在本地响应用户请求,从而极大地缩短请求响应时间。

与上一种分布策略相反,如果将缓存分布在靠近 Web 服务器的地方(如 reverse proxy caching 与 push caching 系统中的作法),往往可以改善服务器的吞吐率等性能。这一类缓存分布策略对于那些对延迟非常敏感的内容(如音频和视频信息)来说,是非常关键的。将缓存分布于靠近或代表内容提供商的地方,可使提供商改善内容的可扩展性和可获得性,但显然,仅对该提供商来说有效,其他内容提供商必须做同样的事。

3 Web Cache 的体系结构

3.1 集中式的 Caching Server 机制

• 代理缓存 代理缓存服务器(proxy cache server)的作用主要是截获来自客户的 HTTP 请求,如果它在缓存中找到了客户所需的对象,则将其返回;否则,就求助于该包含该对象的源 Web 服务器,并代该用户去取来该对象,并可能将其缓存在自己的缓存中,最终将其传送给客户。代理缓存往往位于网络的边缘(如公司、学校里的网关/防火墙、Proxy 服务器),可以服务大量的内部用户。使用代理缓存系统,往往可以节约大量广域网带宽,改善的响应时间,增加静态的、基于 Web 的数据或对象的可获得性等。

• 逆向代理缓存 在代理缓存技术中,有一种变形,即“逆向代理缓存”,就是将缓存部署于靠近信息内容源头的地方,而不是靠近客户。这种解决方案对于那些访问量很大、又希望保证有较高 QoS 水平的 Web 服务器来说,是很有吸引力的。在支持 Web hosting farms(将多个虚拟域映射到一个物理站点上)方面,逆向代理缓存也是一种很有用的机制。前者是一种面向 ISP 的越来越常见的服务。注意,逆向代理缓存

* 本文的工作得到了国家 863 项目“应用服务器的运行、管理及调度技术”(项目编号:2001AA113050)及江苏省高技术产业化推进项目“基于多服务器群的大规模多媒体信息服务平台的研究和开发”的资助。顾铁成 博士研究生,杨书慧 硕士研究生,陆桑璐 副教授,陈道蕃 教授,博士生导师,研究方向为分布与并行计算。

是完全独立于客户端的代理缓存的。这两种缓存机制可以共存,并联合起来改善从用户、网络、服务器等的角度所感知到的 Web 性能。逆向代理缓存分布于靠近 Web 服务器的地方,用于提升 Web 服务器的可扩展性、可用性、容错性,也可以起到负载均衡器的作用。

·透明代理缓存 消除了代理服务器需要用户手工配置浏览器这一不足。透明缓存的工作机制是这样的,它截获 HTTP 请求(即目标端口地址为 80 的 TCP 通信),并将它们重定向至 Web 缓存服务器或缓存群(caches cluster)。透明代理缓存的长处同时也是其主要弱项:它不能保持连接的端点固定,这对于有些要求在连续请求间(或涉及多个对象的一个逻辑请求中)能够保持状态的应用来说,就是一个大问题了。另外,对所有外发的 HTTP 请求进行过滤也会增加额外的延迟^[1]。它有两种部署方式:部署在交换机一级,或在路由器一级。

·下推式缓存(Push caching) 其主要思想是尽可能地使缓存的数据接近真正需要、请求那些信息的客户^[2]。当源 Web 服务器发现了请求来自何处后,就可以对数据进行动态的镜像。例如,如果发现由于来自南方的请求增加的缘故,对北方的某一个网站的通信量突然增加,那么该北方网站的反应就可以是在南方启动一个(下推式)缓存系统。下推式缓存主要是面向内容提供商的,他们能够控制可能部署缓存的潜在场点。

3.2 分布式的 Web Caching Server 机制

单个缓存代理服务器存在着一些问题,如易形成性能瓶颈和单个故障点,性能和可扩展性都受到影响,最终反而会降低缓存的有效性。因此,可采用分布式(或称协同式的)Web Caching 技术和系统,即让一组缓存服务器协同工作,来服务一群客户,从而使整个 Web cache 系统在负载均衡、容错等方面都具有较好的性能。协同式 Web Caching 的思想与技术源自于传统的分布式文件系统,其构成形式有层次型、分布式和基于 cluster 的等几种:

·层次型的协同 Web 缓存 层次型协同 Web 缓存技术最早见于 Harvest 系统,它是由 ARPA 资助的一个项目,是第一个层次型 Web cache 系统。它后来演化出两个系统:Squid 和 NetCache,前者已成为了事实上的 Web Cache 标准。

·分布式缓存 分布式 Web 缓存具有这样一些特点,即代理共享;它试图使多个缓存服务器共享各自的缓存,以形成一个更大的、统一的分布式缓存。

·基于群的、利用组播技术的缓存:还有一种协同 Web 缓存的组成形式,即将多个缓存服务器形成群,并利用组播(multicast)技术在它们之间进行通信,这类缓存可以分为自适应的 Web caching、动态 Web caching、LSAM 等。

4 Web Cache 间的通信机制

分布式的 Web caching 系统在可扩展性、可用性等方面都较好,也可以充分利用各个缓存在物理上的本地性。在可扩展性与可用性方面,采用多个分布式的缓存,使得一个系统可以处理大量的并发客户请求,同时可以抗得住正常操作中某些缓存的失败。在物理局部性方面,在带宽一定的情况下,使缓存靠近特定的用户群可能是一种有效的降低平均网络延迟的方法,因为在一个用户的位置与他所请求的对象之间往往有某种联系。

在分布式缓存系统中,通过将所缓存的对象分布在各个

缓存之间,可以实现负载均衡,此外,缓存之间通过互相查询、互相通信,使得整个分布式缓存系统可以有效地在内部解决用户请求。

有五种较为著名的缓存间通信协议^[3]:ICP,cache digests,CRP,CARP,以及 WCCP。在这些协议中,ICP 的历史最长,也是最成熟的。它是从 Harvest 中的缓存通信机制发展而来的,在 Squid 中得到了进一步的探索。在 ICP 机制中,一个缓存可以向其他缓存发出查询,以确定可以检索到客户所请求对象的最佳缓存位置。ICP 模型包含了一种“请求-回答”(request-reply)模式,它通常用 UDP 上的单播技术来实现。

分布式缓存可以改善缓存系统的可扩展性,但有时也会阻碍它,如 ICP 后来所揭示的。文[4]提出了缓存系统的理想深度。例如,深度大于四层的树会导致较为明显的延迟。另一个可扩展性方面的问题是 ICP 消息的数量,它随着缓存数量的增加而急剧地增加。正如文[5]中所指出的,在同级缓存的数量与所发送的 ICP 消息数量之间有着直接的关系。

关于 ICP 的研究还带出了组播问题,它也是 adaptive caching design 的一项关键技术。其 CRP 协议利用了组播技术来查询缓存网络(cache meshes)。重迭的节点被用来将未决的查询转发至拓扑中的其他缓存网络中。为了优化网络中查询路径,自适应缓存技术采用了 CRP 来决定内容起源的可能方向。虽然组播是在查询同级缓存时所建议的与 ICP 搭配使用的解决方案,这种做法的可扩展性如何仍然是不明确的。

与缓存间通信有关的另一项技术是所谓的“cache digests”(缓存对象信息内容摘要)概念,如 Squid^[5]和 Summary Cache^[6]中所实现的那样。摘要技术通过对同级缓存中所包含对象的情况进行总结,即可以减少缓存间通信的量。于是,请求转发可以变得更为智能和更为有效。这与 URL 路由表的思想非常相似,后者在自适应缓存技术中用到了,也是用作为一种更加智能地转发请求的方法。

5 多媒体代理缓存服务器系统

随着因特网的迅猛发展和数字视频技术的成熟,出现了各种基于因特网的流媒体视频应用,如网上视频点播、视频会议、远程教学、数字图书馆等。在这些应用系统中,视频对象的传输具有实时性强、数据量大等特点,对网络带宽要求很高。这样一来,视频服务系统的规模、可扩展性等往往就受到了限制。解决这一问题的办法一般有两种,即采用视频组播(video multicasting)和 Web Caching 技术。后者的目的是尽可能将流媒体的传输问题推向网络的边缘。

但是,在采用 Web Caching 技术来部分地解决视频对象的传输问题时,会遇到一些新的问题。由于视频对象的尺寸往往都较大、生存周期较长,如果将其作为一个整体来进行缓存的话,在 Web Cache 系统中缓存的视频对象数量较少,会导致命中率低、置换不易且开销较大、传统的置换策略不适用等问题。因此,研究人员提出了一系列的多媒体代理缓存技术,其主要做法都是都对视频对象进行部分缓存。亦即,将视频对象划分为几个部分,以部分为单位来进行缓存。这方面的技术有 Video Staging^[8]、Prefix Caching^[9]等。文[10]还研究了分层编码的视频对象通过 Web cache 系统进行缓存及自适应传输的问题。

小结 Web Caching 是解决因特网基础结构可扩展性的一种有效的解决技术。本文研究了 Web Caching 技术的内容与作用,分析总结了 Web Caching 研究的一些关键问题,如种类、分布方式和体系结构、Web Cache 间通信协议、多媒体

(下转第 30 页)

$$INFER(x \rightarrow y) = \frac{H(y) - H_x(y)}{H(y)}$$

$H(y)$ 表示 y 这个信息的不确定性,也就是说,推论出 y 这个信息的不可能性的概率是多少。 $H_x(y)$ 表示在已知 x 信息的条件下 y 的不确定性。那么我们可以看出, $INFER(x \rightarrow y)$ 表示 x 和 y 之间的相关性,这个值在 0 到 1 之间。如果这个值等于 0,表示我们不可能从 x 的信息中推论出任何东西来。如果这个值等于 1 表示给定 x 就能推出 y 。

这个公式还是很有用的,因为它告诉我们推论不是一个绝对的问题,而是一个相对的问题。但是,它存在两个很大的问题:①我们非常难以——尽管不是没有可能——决定 $H_x(y)$ 的值;②我们也没有考虑做出推论的计算复杂度。

为了说明第二点,我们可以举如下的例子:通过尝试所有的密钥,我们可以从密文中推导出明文,也就是 $H_x(y)=1$,这个地方 x 表示明文, y 表示密文。但是,实际情况是,我们很难这样做。有关推论控制的详细讨论可参见文[15]。

6 加密控制

一般的数据库管理系统直接将数据以原始的形式存储在系统中,如果一个知道数据文件格式的专家有意的进行攻击,后果不堪设想。理想的情况是对数据进行加密存储。

数据库数据加密有它自身的特点。首先,它不能像一般数据那样,以整个数据文件为单位,从头到尾进行加密。数据库操作的特点要求可以从数据文件中抽取任何记录。也就是,加密必须解决随机的从数据库文件中某一段数据开始解密的问题。同时,加密、解密是数据调入调出的必由之路。因此,低效的加、解密算法是不能满足数据库加密要求的。另外,数据库数据一般来说会存储无限长的时间,而数据库数据又是非常的大,如果要更改密钥,代价是很大的。这就导致了密钥时间上保持很久,增加了泄密的可能性。索引字段不能加密,因为在非明文状态下,根本不可能进行索引。也不能进行关系比较,因为被比较的字段 SQL 语句无法辨认。SQL 语言中的函数将对加密数据失去作用。

因此,总的来说,数据库加密存在很多的问题。一般的解决方案是仅仅是对数据库进行部分的加密,也就是对机密的信息进行加密。有关加密控制的详细讨论,可参见文[17]。

结论 数据库管理系统的安全研究取得了不错的进展,但是,应注意的是,有些问题还需要进一步的研究和讨论。

事务方面,目前很多的事务研究还仅仅是局限于严格的 ACID 事务要求。随着一些高级应用,如办公自动化,CAD/

CAM, WEB 服务等,对数据库的要求,事务模型需要加以扩充。在新一代数据库方面的研究相对来说就更少了。对面向对象数据库来说,现在还缺乏一个一般的模型来描述存取控制。各种各样的模型虽然提出了很多,但是,如何验证这些模型的正确性,却成了一个问题。验证数据库管理模型是否安全也许是一个研究的方向。目前,对审计的研究也相对较少,如何使用审计来进行入侵检测,特别是在数据库中,还有待考虑。

参考文献

- 1 Sandhu R S, Samarati P. Access Control: Principles and Practice. IEEE Communications, vol. 32, no. 9, pp. 40~48[SS94]
- 2 Denning D E. Cryptography and Data Security. Addison-Wesley Publishing Company, 1983
- 3 Fernandez E B, Summers R C, Wood C. Database Security and Integrity. Addison-Wesley Publishing Company, 1981
- 4 Lampson W. A Note on the Confinement Problem. Comm. of ACM, Oct. 1973
- 5 National Computer Security Center. Polyinstantiation Issues in Multilevel Secure Database Management Systems: [NCSC Technical Report-005]. Volume 3/5, May 1996
- 6 Sandhu R S, Coyne E J, Feinstein H L, Youman C E. Role-Based Access Control Models. IEEE Computer, 1996, 29(2): 38~47
- 7 Denning D E, et al. The SeaView Security Model. In: Proc. IEEE Symposium on Security and Privacy, 1998. 218~233
- 8 Sandhu R, Chen Fang. The Multilevel Relational (MLR) Data Model. George Mason University, 1996
- 9 Baraani-Dastijerdi A, Pieprzyk J, Safavi-Naini R. Security In Databases: A Survey Study. [TR-96-02]. Department of Computer Science, The University of Wollongong, Wollongong, Australia, 1996
- 10 Harrison M, Ruzzo W, Ullman J. Protection in operating systems. Communications of the ACM, 1976, 19(8): 461~471
- 11 Denning D E. Database Security. Annual Reviews Inc., 1988, 3: 1~22
- 12 Biba K. Integrity Considerations for Secure Computer System. [ESD-TR-76-372]. Hanscom AFB, Mass. Air Force Electronic System Division, 1977
- 13 Smith K. Execution Ordering for Multilevel Secure Rules. In: 4th Int. Workshop on Issues in Data Engineering, RIDE-ADS, Houston, TX, Feb. 1994. 98~104
- 14 National Computer Security Center. Inference and Aggregation Issues in Secure Database Management Systems: [NCSC Technical Report-005]. Volume 1/5, May 1990
- 15 Jajodia S, et al. Multilevel Secure Transaction Processing: Status and Prospects. In Database Security X: Status and Prospects, Chapman & Hall, London, 1997. 79~98
- 16 刘启原, 刘怡编著. 数据库与信息系统安全. 科学出版社, 2000
- 17 Rabitti F, Bertino E, Kim W, Woelk D. A Model of Authorization for Next-Generation Database Systems. ACM Transactions on Database Systems, 1991, 16(1): 88~131

(上接第 63 页)

代理缓存技术等。有关 Web Caching 尚存在着一些未解决问题,如动态和实时数据的缓存问题、影响置换算法性能的一些因素、Caching 对动态文档的作用、置换算法与检查文档一致性的算法之间的关系问题、多媒体代理缓存的特点等问题。针对这些问题,还可以做进一步的研究。

参考文献

- 1 Danzig P. Network Appliance NetCache White Paper: Architecture and Deployment. <http://www.netapp.com>
- 2 Gwertzman J, Seltzer M. The case for geographical pushing caching. Fifth Annual Workshop on Hot Operating Systems, 1995
- 3 Melve I. Inter-cache communication protocols. IETF WREC Working Group Draft, 1999
- 4 Baeentsch M, et al. World web caching - the application level view

- of the Internet. IEEE Communications Magazine, 1997, 35(6)
- 5 Claffy K, Wessels D. ICP and the Squid Web Cache. 1997
- 6 Fan L, et al. Summary cache: A scalable wide-area web cache sharing protocol. Computer Sciences Dept. Univ. of Wisconsin Madison, 1998
- 7 Aggarwal C, Wolf J L, Yu P S. Caching on the World Wide Web. IEEE Trans. on Knowledge and Data Engineering, 1999, 11(1)
- 8 Zhang Z, et al. Video-Staging: A Proxy-Server-Based Approach to End-to-End Video Delivery over Wide-Area Networks. IEEE/ACM Trans. On Networking, 2000, 8(4)
- 9 Sen S, Rexford J, Towsley D. Proxy Prefix Caching for Multimedia Streams. IEEE Infocom 99
- 10 Rejaie R, Kangasharju J. Mocha: A Quality Adaptive Multimedia Proxy Cache for Internet Streaming. Proceedings of NOSS-DAV, Port Jefferson, New York, June 2001