

基于混合索引的浏览器缓存协作系统

于 卿, 古志民, 程慧芳

(北京理工大学计算机学院, 北京 100081)

摘 要: 构建了一种基于混合索引的浏览器缓存协作系统, 它充分利用客户端浏览器缓存的内容, 以 P2P 方式交换 Web Cache, 克服了集中式代理服务器存在的一些问题。客户端的 Internet 访问响应时间是系统设计的主要依据。索引技术是系统中影响响应时间的关键技术, 为了提高响应时间, 对索引技术进行了讨论, 并针对该应用的特点进行了改进。系统具有以下优点: 易于扩展, 网络资源开销小, 响应时间短, 索引更新及时。

关键词: Peer-to-Peer; Web cache; 索引

Browser Cache Collaboration System Based on Hybrid Index

YU Qing, GU Zhimin, CHENG Huifang

(School of Computer Science and Engineering, Beijing Institute of Technology, Beijing 100081)

【Abstract】In this paper, a Web browser cache collaboration system is built based on hybrid index method in hybrid P2P network. By fully utilizing the cached pages stored in Web browser and exchanging cached pages, the system avoids the scalability problem of traditional proxy system. The main propose of the system is reducing the Internet accessing latency in large scale Intranet. The index technology, which is the key component affects the latency, is revised for this system in order to get better efficiency. The system has the following traits: scalability, low network cost, low latency, index updating in time.

【Key words】 Peer-to-Peer(P2P); Web cache; Index

1 概述

当前Internet的接入方式一般是: 客户接入内部网(Intranet), 内部网经一个出口接入Internet。内部网内的网络连接是快速可靠廉价的, 而对Internet的连接是慢速不可靠的, 且是收费的。文献[1]提出基于Peer-to-Peer网络[2]的Web Cache系统, 它利用浏览器缓存, 通过P2P网络共享各个节点的浏览器缓存内容。它是一个大规模分布式Web Cache系统, 克服了利用代理服务器[3]构建的Web Cache系统的缺点, 例如需要超大存储空间、扩展性差、单点失败等。文献[4]提出一个基于混合式P2P网络的Web Cache系统。系统的索引服务器存储各个节点的缓存索引信息, 它还充当代理的职责。此文对P2P Web Cache系统给出一个整体的可行的框架, 但对如何建立高效索引及其更新, 如何进行对等通信等问题没有给出明确分析。文献[5]提出一种基于Pastry网络的Web Cache系统。该文利用Pastry网络, 将所有的缓存分布在各个节点上进行统一管理。此文没有考虑路由开销以及响应时间等问题。文献[6]提出了一种基于地理位置的路由算法, 并对路由范围进行了讨论。结论是在一个部门内或一栋大楼内进行flooding效率和命中率都较高。此系统是纯P2P网络, 其路由算法的实质仍是flooding方式, 只不过对其范围有所限定。文献[7]提出了一种基于混合式P2P网络的Web Cache系统。系统采用胖节点的方式, 胖节点负责某个区域的节点管理、索引等问题。本文也未对索引管理等问题进行讨论。

本文在文献[7]的基础之上, 详细讨论在一个区域内的系统工作流程和各个模块的详细设计, 并对系统的核心部分(索引)进行详细讨论, 分析了现有的技术, 并结合本系统的特点对索引进行了改进, 对其性能进行了理论分析。

2 系统详细设计

体系结构如图 1 所示。在某个区域内部采用索引 P2P 网络, 节点资源由胖节点管理。它保存着该区域内所有节点的地址和节点所共享缓存内容的索引。进行查询时, 系统先在本区域胖节点处查询, 如未找到胖节点向其它胖节点转发查询请求。每个节点有自己唯一的标识, 即 PeerID, 由 P2P Register 统一分配。

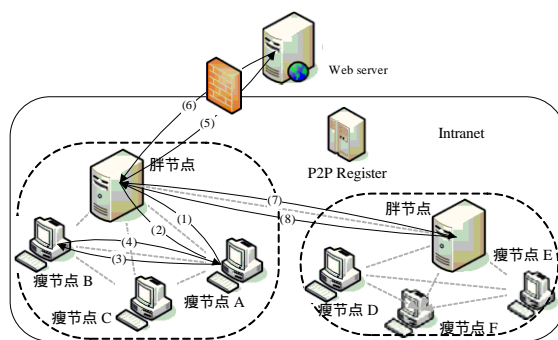


图 1 混合式网络缓存结构

系统工作流程如下: (1) 一个节点发出 HTTP 请求, 客户 A 向胖节点发出查找消息。胖节点查找索引, 并根据一定的调度策略选择一个缓存了所需页面的最好的节点; (2) 将查找结果返回给 A; (3) A 根据返回结果向节点 B 发送文件请求;

基金项目: 北京理工大学基础研究基金资助项目(0301F18)

作者简介: 于 卿(1981-), 男, 硕士生, 主研方向: 计算机体系结构; 古志民, 教授; 程慧芳, 博士生

收稿日期: 2006-03-06 E-mail: qingy@bit.edu.cn

(4)B 从本地缓存中找到该网页并返回给 A。如果此时 B 的缓存已经被淘汰了,则 B 向 A 返回失败标志,同时向胖节点报告结果。如果胖节点没有找到,则它向其它胖节点询问;(5)如果系统不能找到,则 A 向 Web 服务器(Internet)发出请求;(6)从原服务器得到网页。

系统的模块见图 2。在此我们只分析在某一个区域内的情况。这些讨论结果也可以用于整个网络。在瘦节点处有 3 个模块:缓存收集模块(cache collector),缓存提供模块(cache provider),http 处理模块(http handler)。胖节点除了运行普通节点的程序,还要额外运行一个索引服务程序。它包括两个模块:Peer 管理器(peer manager)和索引管理器(index manager)。

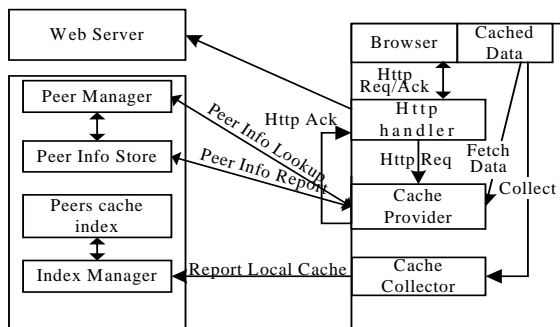


图 2 系统模块

缓存收集模块:负责定期收集本地的浏览器缓存列表,以增量形式汇报给胖节点。它直接利用浏览器的缓存作为节点端缓存,这就避免了处理缓存失效、替换算法等问题,这些问题由浏览器本身处理。模块内部保存有浏览器缓存的列表,它每隔一定时间取得一次浏览器的最新缓存列表,与上一次缓存列表相比较,将增加和替换的 URL 发送给胖节点。此模块还负责 Peer 加入和离开系统的处理。Peer 加入系统时,此模块将本地的所有缓存对象的列表发送给胖节点。

缓存提供模块:负责和其它瘦节点通信。它响应其它节点发出的 HTTP 请求。收到请求后在本地浏览器缓存中寻找请求的缓存对象。如果此网页存在并且没有失效,则直接返回给请求此网页的 Peer。否则返回失败消息并向索引服务器汇报。此模块还负责向胖节点汇报 Peer 状态等。

HTTP 请求处理模块:负责和本地的浏览器通信。它截获浏览器的请求,将 HTTP 请求转发给胖节点,根据胖节点返回的结果,向命中的 Peer 请求缓存的文件,并封装成 HTTP 响应的格式,返回给浏览器。系统借用代理服务器技术来实现 http 请求的截获功能,在浏览器中将代理服务器设为本地即可,系统对浏览器是透明的。

Peer 管理器负责维护 Peer 的状态,处理 Peer 加入和离开等事件。Peer 加入系统和离开系统时,需要申请和释放资源。为了避免频繁的资源申请,我们采用资源池的方法。系统记录 Peer 的最近响应时间,用来维护 Peer 的健康状况,Peer 每隔一定时间向胖节点发送一个 Keep-alive 消息,胖节点收到此消息后更改最近响应时间。当某个 Peer 超过一定时间未响应,胖节点认为此 Peer 失效,从系统中将此 Peer 信息清除。索引管理负责处理查询消息,它还负责管理本地索引。而索引机制和更新策略是影响本系统性能的关键部分,系统要求在较低虚假命中率的前提下,尽量使索引的查找和更新等操作效率较高。

3 主要实现技术

3.1 更新策略

每个节点定期向胖节点汇报浏览器缓存的变化情况(更新)。更新策略影响系统的性能、虚假命中等问题。为了不产生虚假命中,应该对浏览器缓存进行实时监控,但这样对节点和胖节点来说开销都很大。可以观察到其变化是有规律的,见图 3。一般的用户浏览情况:打开一个网页,浏览一段时间(大概 3min ~ 5min),再打开一个网页再浏览,如此反复。每打开一个网页,缓存数目大概变化 20~30 条(新增和淘汰)。据此可以决定节点的更新策略:每隔 4min 更新一次。这种策略可能引起虚假命中,最差的情况是节点刚刚汇报后用户立即访问网页,此时约有 25 条记录没有更新。最理想情况是浏览器内容更新后立即到达汇报时间,此时没有虚假命中。

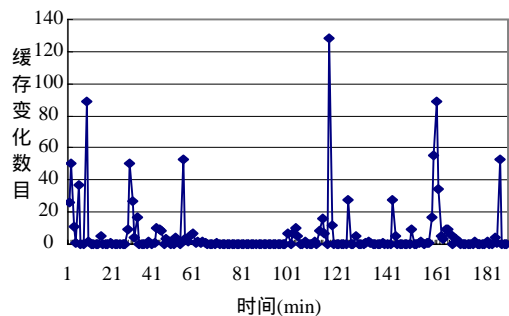


图 3 典型浏览器缓存变化状况

3.2 索引策略

为了进行快速的 URL 比较,我们将 URL 通过 MD5 散列算法转变为 128 位整数进行存储和操作。索引表中的索引项为 URL 对象的 MD5,每个索引项的内容为缓存此网页的 PeerID,这些 PeerID 以链表形式存储,最新加入的排在链表的最前端,见图 4。在查找时直接取得第 1 个节点即可,它所缓存的内容是最新的,而且此节点替换此网页的可能性最小。索引表项的存储方法有两种:(1)随机存储法。所有的 MD5 值放到一个链表中,不进行排序。更新时直接添加到链表尾部,更新速度快,查找效率极低。这种方法适用于更新频繁而查找要求性能不高的系统。(2)顺序存储法。所有的索引项存储在一个大的数组中,数组是按顺序存储的。查找效率高,更新效率较低。这种方法适用于查找非常频繁而更新操作很少的应用。

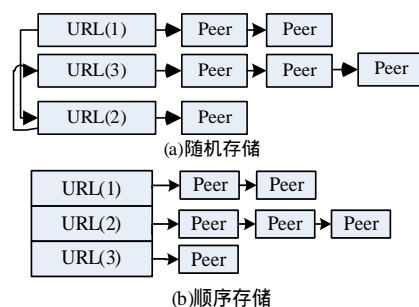


图 4 索引存储方式

上述两种存储方法分别适用于频繁更新和频繁查找的应用,在本系统中应用上述两种方法效率都不高。因此我们将两种索引方法结合。首先有一个顺序存储的索引,作为主要的索引存储方法(称为主表),它的索引表是只读的,这样避免了更新慢的问题。建立一个随机存储表,作为临时更新表。只有在主表中查找不到时,才在临时表中操作。所有新增的索引项都直接插入临时表。当临时表增长到一定长度时,查

找速度就会变慢，可以将临时表并入到主表中。这个操作分为两步：临时表的排序和两个顺序表的合并，此操作可以在后台进行，不影响查找和更新操作。假设系统中所有 Peer 总共有 w 个索引项，设主表长 l ，临时表长 k ，则在选定合适的 l 与 k 的情况下（一般 $l \ll w$ ），此种方法会优于前两种方法。几种方法的比较见表 1。对改进的索引的操作算法如下：

查找流程：

- (1)在主表中查找，采用折半查找法，如果找到，跳转(3)；
- (2)在临时表中蛮力搜索，若找到跳转到(3)，否则返回查找失败；
- (3)从找到索引项中取得最新一个 Peer 返回。

插入流程：

- (1)查找这个 URL，如找到转(3)；
- (2)创建一个新的索引项添加到临时表的最后；
- (3)在找到的或创建的索引项中，将 Peer 插入链表。

删除流程：

- (1)查找这个 URL，如未找到，返回失败；
- (2)在找到的索引项中，查找 Peer，并从链表中删除，返回成功。

合并流程：

- (1)删除临时表中 Peer 链表为空的索引项；
- (2)将临时表排序；
- (3)将临时表合并到主表中。

表 1 几种索引方法的比较

	随机存储法	顺序存储法	改进的方法
查找	$O(w)$	$O(\log(w))$	$O(\log(l)) + O(k)$
更新	$O(1)$	$O(w)$	$O(\log(l)) + O(k)$
合并	-	-	$O(k \cdot \log(k) + \log(l)) + O(l + k)$
使用范围	更新频繁	查找频繁	更新和查找次数接近

3.3 索引性能分析

对索引的性能进行一些理论上的分析。查找操作时间为 t_s ，系统需要查找主表和临时表。主表采用二分查找， $t_{s1} = \log l$ ；临时表采用蛮力搜索， $t_{s2} = k$ ；总查找时间 $t_s = t_{s1} + t_{s2} = \log l + k$ 。插入和删除操作都要先找到索引项，因此 $t_i = t_d = t_s$ 。合并时间 t_m ，对临时表进行快速排序 $t_{m1} = \frac{1}{2}k \cdot \log(k)$ ，然后将两个有序队列合并 $t_{m2} = \frac{1}{2}k \cdot \log(l)$ ，以及移动时间 $t_{m3} = l + k$ ，则 $t_m = \frac{1}{2}k \cdot \log(l) + \frac{1}{2}k \cdot \log(k) + l + k$ 。

选择适当的临时表长度 k ：在一段运行时间内合并次数 $n = \frac{w}{k}$ ，总的合并时间 $t = n \cdot t_m = w(\frac{1}{2}\log(l) + \frac{1}{2}\log(k) + \frac{l}{k} + 1)$ 。对此公式求最优化，可得最优的 k 。

4 实验与相关分析

对系统进行性能测试，采用模拟与真实节点相结合的方法。试验系统由胖节点、虚拟节点和真实节点通过 100M 网络连接而成。虚拟节点依照用户在浏览网页的一般行为产生多个用户的访问请求，它还可以读取 proxy 的 log 文件，根据真实的访问记录产生多个用户的访问请求。

4.1 索引性能测试

本实验主要测试索引模块的性能，测试在不同负载(用户数目、系统中总的索引数目)情况下，索引查询和更新的性能。

使用代理服务器的日志产生节点的访问请求。日志来自 NASA 2005 年的数据(从 ftp.ircache.net 下载)，总共 7 天，将同一个 IP 看作一个节点。首先根据请求的多少截取数据，测试在不同索引规模(索引项多少)下不同方法的性能，结果见表 2。再根据节点的多少截取数据，测试在不同网络规模下不同方法的性能，结果见表 3。可以看出顺序存储方法更新

时间很长而查找时间较短，随机存储方法更新时间很短而查找时间较长。而我们提出的改进方法更新和查找时间都较短。

表 2 不同请求数目下索引方法的比较

(ms)	顺序存储		随机存储		改进的方法	
	更新	查找	更新	查找	更新	查找
10	25	6	0	1	0.1	0.1
50	39	13	0	10	0.3	1
200	36	10	0	15	0.6	4
500	73	15	0	40	3	14
1000	85	15	-	-	2	26

表 3 不同节点数目下索引方法的比较

(ms)	顺序存储		随机存储		改进的方法	
	更新	查找	更新	查找	更新	查找
节点数						
50	355	18	0.1	1	0.9	0.2
100	274	14	0	50	3	0.9
200	375	12	0	33	3	4
400	173	17	-	-	4	13
700	195	34	-	-	8	40

测试系统在高负载情况下的性能。首先测试在系统索引表较长时的查找性能。在系统运行前，我们在主表中生成 2×10^7 个索引项。虚拟节点根据一般的用户浏览习惯发送查询请求，测试多个节点同时工作时查询的响应时间。结果如图 5 所示：在系统负荷较重时，每个节点的平均查询响应时间仍随节点数目线性变化。当只有一个节点时，响应时间仅在 1ms 以下。当同时工作节点数目较多时，每个节点的平均响应时间仍然不足 1s，是可以接受的。

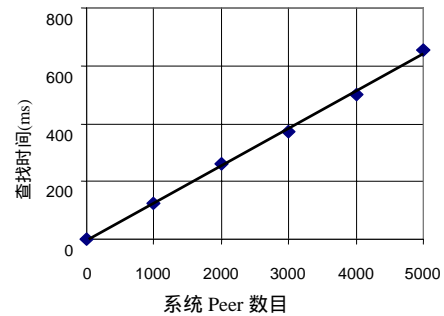


图 5 满负载情况下的查找时间

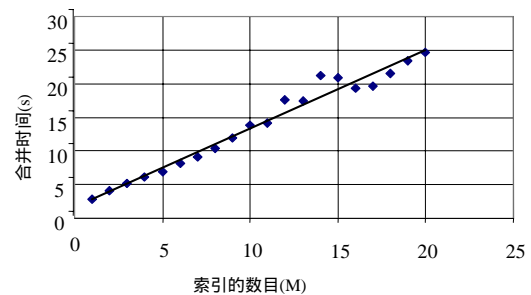


图 6 满负载情况下的合并时间

其次对系统在节点较多(5 000 个)时的性能进行测试。本实验主要测试系统合并时间。系统初始时索引表为空，虚拟节点产生 5 000 个节点，并根据一般的用户浏览习惯产生浏览器 Cache。随着节点不断地发送添加消息，索引表会不断增长，达到一定长度后系统会对索引进行合并操作。我们测试索引长度不同时，合并操作的时间。测试结果见图 6，可见合并时间随系统规模呈线性增长，当系统的索引项较多时，合并时间也不超过 25s。

4.2 真实性能测试

本实验主要以客户端响应时间为测试目标,测试在真实环境下,系统的整体响应时间,即用户得到一个网页的时间。我们要将此时间与直接从服务器端得到的时间相比较。所有测试都是在服务器节点满负荷情况下进行的测试。测试结果见图7,可见相对于直接访问 Internet 的方式,本系统的响应时间与网页大小无关(或影响很小),且系统的总体响应时间相对直接访问方式要低很多。同时可以看出,本系统对于较大的网页有更好的性能。

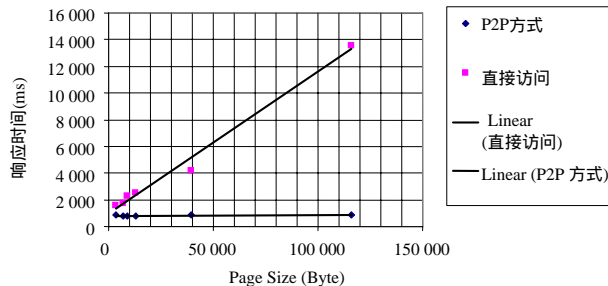


图7 真实性能测试

5 结论及后续工作

本文提出了一种基于混合式 P2P 网络的 Web Cache 系统,它能很好地解决在内部网中使用单一代理服务器所带来的一些问题,对其体系结构和模块进行了详细设计。提出适用于本应用的高效索引方法,并对其性能进行分析。测试表明,此索引方法能够高效地完成查找、更新操作,满足系统要求。我们的系统比直接访问 Internet 效率要高,响应时间

大大缩短,特别是响应时间与文件大小几乎无关。此文只考虑在整个 P2P 网络中的一个区域内的问题,而构建在整个 P2P 网络中的系统,还需要在今后的工作中对本系统进行扩展。

参考文献

- 1 Xu Zhichen, Xiao Li, Zhang Xiaodong. Data Integrity and Communication Anonymity in Peer-to-Peer Networks[R]. Hewlett Packard Laboratories, Technical Report: HPL-2001-204, 2001-08.
- 2 张晓东. 网格及对等分布式计算技术(Grid and P2P)[EB/OL]. 2004. <http://www.gridchina.org/>.
- 3 Jia Wang. Survey of Web Caching Schemes for the Internet Computer Communication Review[J]. IEEE Internet Computing, 1999, 29(5).
- 4 Li Xiao, Zhang Xiaodong, Xu Zhichen. On Reliable and Scalable Peer-to-Peer Web Document Sharing[C]/Proc. of International Parallel and Distributed Processing Symposium. 2002.
- 5 Iyer S, Rowstron A, Druschel P. SQUIRREL: A Decentralized, Peer-to-Peer Web Cache[C]/Proceedings of the 21st Annual Symposium on Principles of Distributed Computing. 2002.
- 6 Mao Yonggen, Zhu Zhaoming, Shi Weisong. Peer-to-Peer Web Caching: Hype or Reality?[C]/Proc. of the 10th International Conference on Parallel and Distributed Systems. 2004.
- 7 程慧芳,古志民.一种基于 P2P 的可扩展 Web 缓存方法[J].北京理工大学学报,2005,25(3).

(上接第 98 页)

监测契约的执行,由此获得资源服务质量、GSC 和 GSP 的行为信息。契约履行负责资源调度结束后,修改契约实例登记项、对调度成功/失败的仲裁、实现支付或相应补偿金支付等。

5 小结

基于契约的资源调度首先以形式化的方式明确了 GSC 和 GSP 之间的责任和义务,便于统一的管理和操作,同时为上层网络监测,结算和支付提供了一致的支持;其次,补偿金的纳入使得 GSP 对资源的性能更加关注,使 GSC 获得更好的服务,促进 GSP 以更加诚信的态度提供资源。本文首次将契约应用于网格资源调度,提出了一个结构化的契约模式,完成了契约匹配算法,考虑了契约的安全问题和相关处理,在此基础上研究了基于契约的网格资源管理体系。下一步的工作是实现以契约为基础的网格资源调度模型,并进行相应仿真,来定量评价契约对资源调度的影响和走向应用必须解决的问题。

理论上,契约简化了资源管理以及网格监测。它为资源调度提供了一个强有力的保障,使得网格监测不再需要深入到资源所在的本地系统采集数据,进而,契约提供的资源使用数据更为可信(在资源所在本地系统的数据可能被更改)。就促进 GSP 提供更可靠的资源,并由此推进计算经济市场更加稳定等方面评价需要一个很长的时间才能体现出来。

参考文献

- 1 Kurian A B, Veljkovic I. Reputation-based Grid Resource Selection[C]/Proceedings of the Workshop on Adaptive Grid

Middle-ware, New Orleans, LA, USA. 2003.

- 2 Cornelli F. Choosing Reputable Servents in a P2P Network[C]/Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii. 2002-05.
- 3 Azzedin F, Maheswaran M. Integrating Trust into Grid Resource Management Systems[C]/Proceedings of International Conference on Parallel Processing, Vancouver, BC, Canada. 2002: 47-54.
- 4 马满福,吴健,胡正国,等.网格资源管理中的信誉度模型[J].计算机应用,2005,25(1).
- 5 Bundy A. A Concise Business Guide to Contract Law[M]. Gower Publishing Ltd. UK, 1998.
- 6 Smith J C. The Law of Contract[M]. Sweet & Maxwell Ltd., 1989.
- 7 Akioka S, Muraoka Y. The Markov Model Based Algorithm to Predict Networking Load on the Computational Grid[J]. Journal of Mathematical Modeling and Algorithms, 2003, 2(3).
- 8 Buyya R. Economic-based Distributed Resource Management and Scheduling for Grid Computing[D]. B.E. Mysore Univ. and M.E. Bangalore Univ., 2004-04.
- 9 Li Maosheng, Yang Shoubao. Research on Grid Resource Reliability Model Based on Promise[C]/Proc. of International on Information and Technology. 2005: 310-315
- 10 Shand B, Bacon J. Policies in Accountable Contracts[C]/Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks, California, USA. 2002-06: 80.
- 11 Ma Manfu, Wu Jian. Credit in the Grid Resource Management[C]/Proc. of Grid and Cooperating. 2005-11: 725-731.