

无线局域网的 Web 缓存系统研究 *

田四化 任鑫 金蓓弘

(中国科学院软件研究所软件工程技术中心 北京 100080) (中国科学院研究生院 北京 100080)

摘要 由于无线环境中的网络和手持设备的限制,缓存机制在提高移动系统性能方面发挥着重要作用。本文描述了适用于无线 LAN 的 Web 缓存系统 EasyCache 系统的实现,给出了基于数据挖掘理论的移动主机端缓存数据预取算法 SEDP。模拟实验数据表明 SEDP 算法取得了良好的效果。

关键词 缓存,预取,移动环境

Research on Web Caching System in Wireless LAN

TIAN Si-Hua REN Xin JIN Bei-Hong

(Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

(Graduate School of the Chinese Academy of Sciences, Beijing 100080)

Abstract Caching mechanism play an important role in improving the performance of mobile systems which must face the constraints in handheld devices and mobile network. This paper describes the implementation of EasyCache system, a Web caching system for wireless LAN, and presents the data prefetching algorithm SEDP for mobile host's cache which refers to data mining theory. Results from simulation experiments show the SEDP in EasyCache system gets better effects.

Keywords Caching, Prefetching, Mobile environment

1 引言

随着无线网络的普及和移动设备的广泛应用,越来越多的人习惯于使用手持设备随时随地访问 Internet^[1]。

基于基础设施的移动网络 (Infrastructure-based Mobile Network)^[2] 是移动计算环境中发展最快的一个部分。图 1 给出了这样的一个网络环境。这里,移动无线设备(通称移动主机,简写成 MH)基于 IEEE802.11 标准(WiFi)与访问点(简称 AP)建立无线通信,并通过 AP 连接到有线局域网,从而通过有线局域网上的代理访问 Internet。

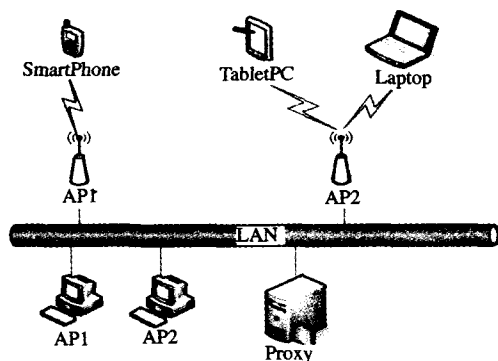


图 1 基于基础设施的移动网络

为了在这样的移动网络下高效地访问 Web 信息,缩短访问响应时间,采用缓存保存用户经常访问的网页文件是一项主要技术,但是无线网络的低带宽、长延迟、容易断线等特点以及手持设备在内存大小和处理能力上的限制,给缓存机制提出了新的挑战。

我们针对移动环境下的网络限制,实现了 EasyWeb 系统,该系统主要包括基于 J2ME 的 Web 浏览器 EasyBrowser 和 Web 缓存系统 EasyCache,为移动用户访问 Web 提供支持。本文给出了 Web 缓存系统 EasyCache 的设计和实现,并详细描述了基于数据挖掘理论的移动主机端缓存数据预取算法及其实验效果。下文按如下方式组织:第 2 节介绍 EasyCache 系统的体系结构,包括系统部署环境、主要组件和系统典型流程,第 3 节详细描述了基于数据挖掘的数据预取算法 SEDP 的设计与实现,第 4 节给出了 SEDP 的系统实验结果,最后全文总结。

2 系统体系结构

Web 缓存系统 EasyCache 在移动主机上存放用户访问过的网页,同时实现网页预取功能,以便能充分利用网络带宽,提高命中率,更好的支持离线访问;另一方面,EasyCache 基于 Web 代理缓存软件 Squid^[3],实现了同一局域网内多个代理上的协作缓存;针对移动主机的移动性,实现了移动主机信息随移动主机切换在代理之间转移,从而有效地支持了代理间的缓存协作。系统还实现了多代理之间的负载平衡,以进一步提高系统吞吐量。

EasyCache 包含移动主机缓存 (MHCache) 组件、Http 代理 (HttpProxy) 组件和代理服务器缓存 (PSCache) 组件,其中, MHCache 组件负责移动主机端的缓存管理; HttpProxy 组件监听代理上的 Http 端口,拦截 Http 请求,解析后将 Http 请求的内容发送给 PSCache 组件,并负责把 Http 应答转发给移动主机; PSCache 组件负责代理上协作缓存的管理。这些组件部署在基于基础设施的移动网络环境下,其中 MH-

*) 本文研究得到国家重点基础研究发展规划 973 资助项目 (编号 2002CB312005) 的资助。田四化 硕士研究生,研究方向:分布式计算,软件工程;任鑫 硕士,研究方向:分布式计算,软件工程;金蓓弘 博士,副研究员,CCF 会员,研究方向:分布式计算,软件工程。

Cache 组件和 Web 浏览器 EasyBrowser 一起部署在移动主机端,HttpProxy 组件和 PSCache 组件部署在代理端,多个代理通过有线局域网互相连接,协同工作,提供对访问 Web 数据的支持。图 2 给出了用 UML 描述的系统部署图。

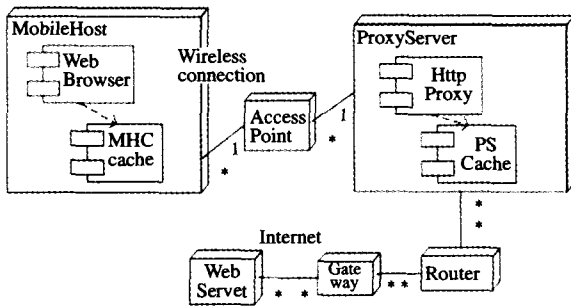


图 2 系统部署图

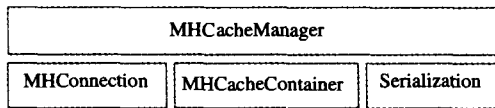


图 3 MHCACHE 组件

2.1 移动主机端组件

MHCACHE 组件负责移动主机端的缓存管理,主要包括下列模块:

- MHCACHEManager 模块,提供访问缓存数据和缓存管理的接口;实现了缓存数据有效性的验证,实现了网页预取功能,提供与缓存代理之间的通讯;并协助代理完成对用户透明的移动切换处理。
- MHCACHEContainer 模块,为 MHCACHEManager 提供缓存数据项的创建、存放、查找、更新、删除等功能,实现缓存替换策略。
- MHConnection 模块,负责移动主机与代理之间基于 Http 协议的消息通信。移动主机在所发出请求的 HTTP 头信息内添加两个信息:所要访问的网页服务器地址和移动主机前一次联系的代理地址,以协助代理之间进行缓存协作。
- Serialization 模块,完成缓存数据的序列化和反序列化。因为 J2ME 的记录管理系统(Record Management System,简称 RMS)只能存储字节数组,所以,对缓存数据进行持久化操作之前,必须将所有对象序列化为字节数组。当读取 RMS 内存储的数据并还原成缓存数据项的时候,需要将字节数组反序列化为对象。

2.2 代理端组件

代理端组件包括 HttpProxy 组件和 PSCache 组件构成。

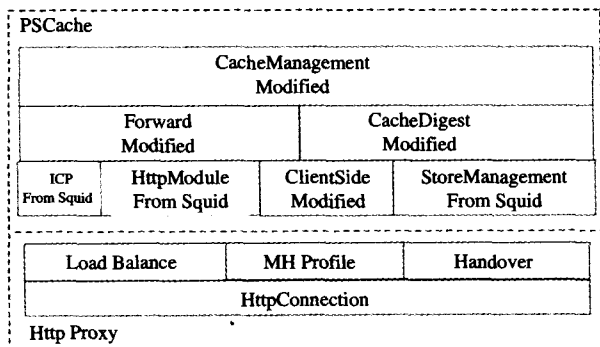


图 4 代理端组件

HttpProxy 组件负责在移动主机和 PSCache 之间进行数据转发,此外,还在 Profile 对象中记录各个移动主机最近访问的 URL,负责切换发生时移动 Profile 并预取 Profile 里面的 URL,协调代理之间的负载均衡等功能。HttpProxy 主要由下面三个模块构成:

- HttpConnection 模块,处理代理与移动主机以及代理之间的 HTTP 通信。解析移动主机发来的 HTTP 请求,获取其要访问的网页地址,返回网页数据。同时获取移动主机上次连接的代理地址,以备与之协作。
- LoadBalance 模块,定期计算本代理 PSCache 组件的负载;当本代理高负载时,广播其状态以通知其它代理更新记录,并选择合适的低负载代理分流用户请求;同时,接收其它代理发来的负载消息,记录低负载代理的 IP 地址。负载的计算综合考虑了代理本身处理能力和负担的访问量,其中,采用代理的 CPU 频率和内存容量的加权和标志其处理能力,单位时间内接收的请求数和当前处理的请求数的加权和标志其当前负担的访问量。
- MHProfile 模块,管理本机所储存的移动主机的 Profile,记录移动主机经常访问的网页的 URL 集合;对比代理 IP 和移动主机上次连接的代理地址 Last-Proxy-IP,如果发生变化,则根据 Last-Proxy-IP 从其它代理上去获取缺失的 Profile。

PSCache 组件缓存经常被访问的网页内容,并响应移动主机发来的 Http 请求。PSCache 是在 Squid 基础上完成的,下列模块扩展了 Squid 的原有功能:

- CacheManagement 模块,修改了原有的调用 ClientSide 模块、Forward 模块、CacheDigest 模块的方式。
- ClientSide 模块,在 Squid 原有的 ClientSide 模块上进行了修改,使得它能与 HttpProxy 的通信。
- Forward 模块,在 Squid 原有的 Forward 模块上进行了修改,将缓存未命中的请求转发给低负载的代理。
- CacheDigest 模块,维护代理所缓存内容的摘要。

2.3 系统流程

用户持有移动主机用 EasyBrowser 访问网页时,调用 MHCACHE 组件获得网页内容;MHCACHE 组件在本地缓存中没有找到所需网页时,通过 WiFi 连接到有线局域网中的访问点;访问点将这个请求转发到代理;代理上的 HttpProxy 组件拦截移动主机发出的 Http 请求,调用 PSCache 组件获得网页内容,并返回给移动主机;如果在代理上没有找到所需网页,那么将通过网关(Gateway)连接到 Internet 上的 Web 服务器,获取目标网页。在获取目标网页后,系统进行 Web 页面的预取工作并进行缓存更新。

3 移动主机端预取缓存的实现

预取是把用户将来可能访问的数据预先存放在本地缓存中。Kuenning 等人^[5]分析了个人在固定时间段内的数据工作集大小、注意力转移次数以及注意力转移所涉及的数据量,这些用户访问习惯的统计结果表明在移动环境中根据用户的访问历史记录预测他们将来可能访问的网页是可行的。

我们提出了一种低开销的网页预取算法 SEDP(Sequence-mining based Easy Data Prefetching)算法^[6],该算法不仅可以有效地提高用户访问 Web 的缓存命中率,而且具有很低的计算开销,尤其适用于计算能力弱的手持移动设备。

SEDP 算法的设计策略如下:

1)根据 Web 访问记录搜索用户经常访问到的网页并放入缓存,设置标志,指示缓存替换策略不把它们替换出去。

2)挖掘用户的访问规律,找出访问网页之后最有可能访问到的网页集合。在下载网页的同时,将与之关联的网页集合预取到缓存中。

令 $\Gamma_i = \{f_1, f_2, \dots, f_j, \dots, f_n\}$ 为一个单位时间(称为一个 session S_i)中用户访问的网页序列,其中 f_i 称作数据项,数据项的集合称为项集。令 $S = \sum S_i$ 。令 $f_i.count$ 为每个 f_i 的出现的次数。令两个数据项 f_x 和 $f_y (x < y)$ 的距离为 $y - x$ 。

令项集 $X (X \subset \Gamma)$ 的支持度(support)为出现 X 的 session 的个数除以 session 总数得到的百分率, minsup 为用户指定的最小支持度。令预取集合的置信度(confidence)代表用户期望的预取成功的概率, minconf 是用户指定的最小置信度。

整个算法分两步。第一步针对策略 1,生成用户访问网页的频繁项集。

算法从 Web 访问日志中,以天为单位,统计用户访问的网页序列,并将数据项支持度大于指定的 minsup 的项,放入频繁项集 F 中。标志这些网页使得在下一一次做数据挖掘之前不被替换出缓存。这里需要对 Web 日志中的所有项循环扫描一遍以计算各网页的访问次数,设 Web 日志的长度为 n ,生成频繁项集的时间复杂度为 $o(n)$ 。我们用数组来存储 Web 日志,用哈希表来存储数据项,则空间复杂度为 $o(n)$ 。

第二步针对策略 2,生成预取集合。

在一个网页序列 $\Gamma_i = \{f_1, f_2, \dots, f_j, \dots, f_n\}$ 中,统计频繁项集中的每个数据项与之距离为 d 的项集 Y ,即对于 $f_i \in S, Y_i = \{f_k | i < k < d + i\}$ 。统计 Y 中项 f_k 出现的次数(这时要过滤掉用户用浏览器的前进、后退功能查看最近浏览过的网页),并设定预取集合 $set_j = \{f_k | f_k.count > f_i.count * minconf\}$ 。在实现上,我们利用了计算频繁项集中的存储 Web 日志的数组和存储项的哈希表。这里要扫描 Web 日志中每一项后面的 d 项,所以时间复杂度为 $o(n * d)$ 。

在用户访问 频繁项集数据项的时候,将根据预取集合进行网页预取。

表 1、2、3 给出了根据 Web 日志计算预取集合的一个例子。

表 1 Web 日志

Session No.	Data Item Requests
1	0 1 2 422 1 2 423 10369 10370 10371
2	0 1 2 422 1 2 1384 1390 1385
3	0 1 2
4	0 1 2 422 1 2 1384 1385
5	0 1 2 422 1 2 1384 9800 8744

表 2 频繁项集

1-itemset	0, 1, 2, 422, 1384
-----------	--------------------

表 3 预取集合

0	1,2,422
1	2,422
2	1,422
422	1,2,1384
1384	1385

表 1 给出的数据是 5 天内的 Web 日志,网页用数字表

示。表 2 给出了应用生成频繁项集的算法步骤得出的频繁项集,其中最小支持度 minsup 设定为 0.6。表 3 给出了应用生成预取集合的算法步骤得出的预取集合,其中距离 d 设定为 3、最小置信度 minconf 设定为 0.6。

当用户访问到网页 422 时,预取操作将查询预取集合,然后将 1、2、1384 这些网页更新到本地缓存中。

4 实验结果

我们采用 Boston University 的 Cunha 提供的用户 Web 访问历史记录(BU Web Trace)^[7]进行实验,它记录了真实的网络请求,从而完整地反映了客户端的访问历史,故被广泛应用于预取算法的实验。在实验之前,我们对原始记录进行了预处理,选择了 session 比较多、访问时间密集的 6 个用户(con1、con6、con10、con19、con62、con206)的数据进行测试。表 4 给出了这 6 个用户的基本信息。

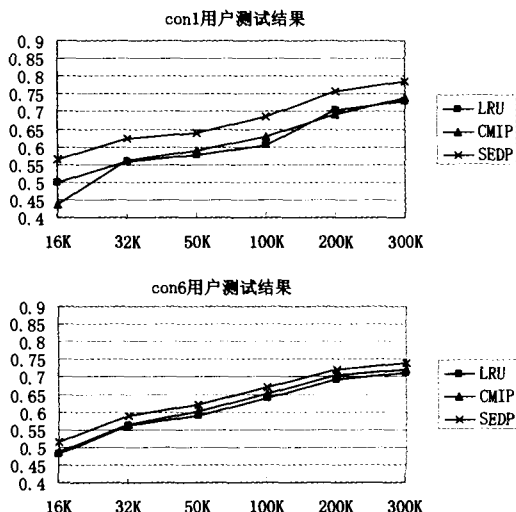
我们将移动主机端用户请求模拟测试程序 Emulator 部署到 JBuilder 9.0 自带的 PDA 模拟器上,将局域网内的一台机器部署上 Tomcat 5.0,作为 Web 服务器,存放用户访问的网页。

表 4 用户的基本信息

用户名	访问过的网页数目
con1	7589
con6	17142
con10	11722
con19	5552
con62	4530
con206	5176

我们比较了下列三种算法的性能:SEDP 算法、CMIP 算法和 LRU 算法。CMIP 算法^[8]是一种基于关联规则的数据预取算法,是从 Web 日志中挖掘访问顺序的关联规则,并利用这些规则预测用户将来可能访问到的数据。该算法中的关联规则更关心网页出现的频率,而不是访问网页的顺序,这样可能使规则关联的两个元素在访问次序上距离很远,从而会导致预取后缓存命中率不提高的情况。

图 5 给出了三种算法 缓存命中率 的测试结果,其中 x 轴表示缓存大小,y 轴表示缓存命中率。



(下转第 285 页)

结束语 本文讨论了有向图的有序二叉决策图存储结构,该存储结构的设计有别于传统的设计思想,它把有向图转化为布尔函数并用基于 OBDD 的图算法对有向图进行操作。基于 OBDD 存储结构的有向图的操作和搜索算法,适合于对大型有向图进行存储和操作。实验表明,该存储结构与传统的邻接表相比,在处理大规模有向图时,具有较高的存储效率。

参考文献

- 1 周海岩. 图的关联链式存储结构. 计算机研究与发展, 1997, 34(10): 101~106
- 2 Akers S B. Binary Decision Diagrams. IEEE Transaction on Computer, 1978, 27(6): 509~516
- 3 Bryant R E. Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. on Computers, 1986, 35(8): 677~691
- 4 Somenzi F. CUDD; CU decision diagram package release 2.3.1. <http://vlsi.Colorado.edu/~fabio/CUDD/cuddIntro.html>, 2001

(上接第 85 页)

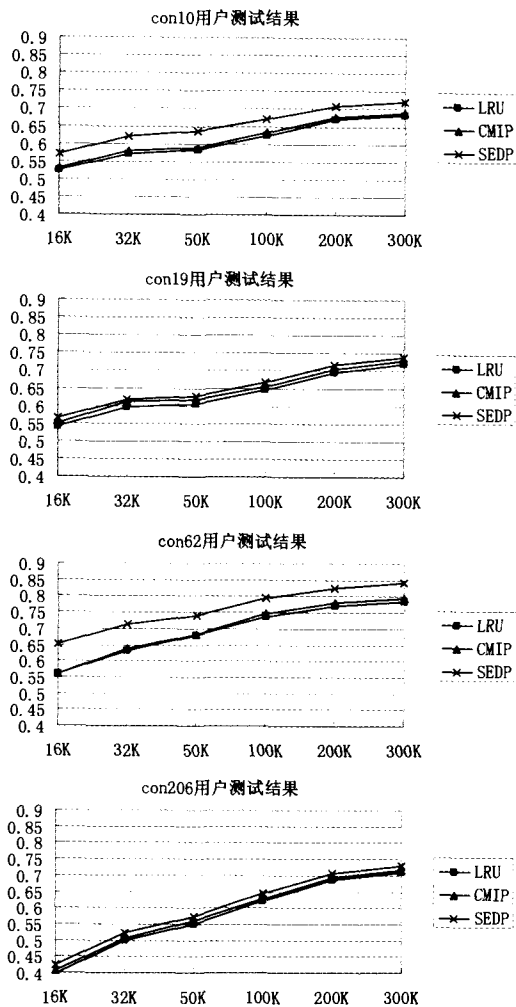


图 5 缓存命中率的测试结果

测试表明, SEDP 算法对于不同用户都能有很好的表现, 其缓存命中率比 CMIP 算法提高 2% 到 6%, 比 LRU 算法提高 2% 到 8%。

表 5 算法执行时间(单位:毫秒)

用户名	CMIP	SEDP
con1	44560.67	446.67
con6	313637.5	2136.5
con10	1091.17	807.83
con19	888.5	270.17
con62	11500.17	828
con206	751	272

表 6 算法占用内存空间(单位:字节)

用户名	CMIP	SEDP
con1	569000	38032
con6	93320	84524
con10	19112	77438
con19	12516	21194
con62	29684	65804
con206	5781	7388

运行上述算法所用时间如表 5 所示,从所用时间上看, SEDP 算法比 CMIP 算法的复杂度更低。运行上述算法所用内存大小如表 6 所示,数据挖掘所用的内存空间并不稳定。但是由于数据挖掘只在缓存命中率低的时候才运行,而且运行频率不高,所以短时间占用较多的内存空间对系统影响不大。

小结 本文主要研究无线局域网内的 Web 缓存技术,我们给出了适用于无线 LAN 的 Web 缓存系统 EasyCache 的实现,该缓存系统能支持手持设备的移动切换,并具有局域网内的缓存协作和负载平衡功能。我们在 EasyCache 中提出了一种基于数据挖掘的预取算法 SEDP,实验证明:SEDP 算法的缓存命中率比 CMIP 提高了 2%~6%,比 LRU 算法提高 2%~8%,而且 SEDP 算法的计算复杂度较低,适用于计算能力较低的移动手持设备。

参考文献

- 1 Mallick M. Mobile and Wireless Design Essentials, Wiley, 2003
- 2 Coulouris G, Dollimore J, Kindberg T. Distributed Systems Concepts and Design, Fourth edition, 2005
- 3 Wessels D, Claffy K. ICP and the Squid Web Cache. IEEE Journal on Selected Areas in Communications, 1998, 16(3)
- 4 Fan Li, Cao Pei, Almeida J, Broder A Z. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. IEEE/ACM Transactions on Networking, 2000, 8(3)
- 5 Kuenning G, Popek G, Reiher P. An Analysis of Trace Data for Predicative File Caching in Mobile Computing. In: Proc. of the 1994 USENIX Conference, June 1994, 291~306
- 6 BU Web Trace. <http://ita.ee.lbl.gov/html/contrib/BU-Web-Client.html>
- 7 任鑫. 无线局域网的 Web 缓存研究与实现:[硕士论文]. 2006
- 8 Song Hui, Cao Guohong. Cache-Miss-Initiated Prefetch in Mobile Environments. Computer Communications, 2005, 28(7): 741~753